Politechnika Warszawska
Warsaw University of Technology
https://repo.pw.edu.pl

Rodzaj dyplomu / Diploma type	Rozprawa doktorska / PhD thesis
Autor / Author	Karolak Juliusz
Tytuł / Title	Metoda przygotowania specyfikacji interfejsu w powiązaniach komputerowych systemów sterowania ruchem kolejowym / The method for the preparation of interface specifications in the interconnection of computerized railway traffic control systems
Rok powstania / Year of creation	2023
Promotor / Supervisor	Zboiński Krzysztof
Jednostka dyplomująca / Certifying unit	Wydział Transportu / Faculty of Transport
Adres publikacji w Repozytorium URL / Publication address in Repository	https://repo.pw.edu.pl/info/phd/WUTa37e0de5e9dd499bbb67574458d2916a/
Data opublikowania w Repozytorium / Deposited in Repository on	Jun 15, 2023

POLITECHNIKA WARSZAWSKA

DYSCYPLINA NAUKOWA – INŻYNIERIA LĄDOWA, GEODEZJA

I TRANSPORT

DZIEDZINA NAUK – NAUKI INŻYNIERYJNO-TECHNICZNE

Rozprawa doktorska

mgr inż. Juliusz Karolak

Metoda przygotowania specyfikacji interfejsu w powiązaniach komputerowych systemów sterowania ruchem kolejowym

Promotor:

prof. dr hab. inż. Krzysztof Zboiński

Promotor pomocniczy:

dr inż. Andrzej Kochan

WARSZAWA 2023

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Składam serdeczne podziękowania

wszystkim, dzięki którym niniejsza praca powstała, a szczególnie:

prof. dr hab. inż. Krzysztofowi Zboińskiemu

za merytoryczne i cenne rady, poświęcony czas,

ogrom życzliwości i wyrozumiałości,

wszelką pomoc i wsparcie

dr inż. Andrzejowi Kochanowi

za współpracę, przekazaną wiedzę i doświadczenie

oraz cenne uwagi i wsparcie merytoryczne

dr hab. inż. Wiktorowi B. Daszczukowi, prof. uczelni

pod którego opieką stawiałem pierwsze kroki w weryfikacji modelowej,

co wpłynęło na ogólny kształt oraz zakres pracy

Katarzynie – mojej kochanej żonie

bez motywacji której niniejsza praca nie powstałaby

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Streszczenie

„Metoda przygotowania specyfikacji interfejsu w powiązaniach komputerowych systemów sterowania ruchem kolejowym”

W pracy przedstawiono metodę pozwalającą wykazać poprawność działania i współdziałania przekaźnikowych i komputerowych urządzeń sterowania ruchem kolejowym (srk), lub wykryć błędy w przypadku ich niepoprawnej konstrukcji.

Rozważona została warstwa logiczna interfejsu urządzeń komputerowych służącego do powiązania tych urządzeń z innymi urządzeniami np. przekaźnikowymi. Metoda objęła opracowanie modeli współpracujących urządzeń uwzględniających ich cechy istotne z punktu widzenia rozważanej warstwy i potencjalnych usterek elementów tych urządzeń.

W przedstawionej w pracy metodologii udało się skutecznie zamodelować przekaźnikowe układy srk i ich powiązania z urządzeniami komputerowymi z wykorzystaniem formalizmu IMDS (Integrated Model of Distributed Systems). Modelami objęte zostały elementy podstawowe dla powiązań: zestyki, cewki, kotwice, połączenia galwaniczne. Modele zostały wykonane w postaci grafów i kodu w języku programowania IMDS. Uwzględniono przekaźniki bistabilne. Modele podstawowe zostały połączone na podstawie zestawień połączeń osiąganych poprzez projekt układu będącego przedmiotem modelowania opracowany w edytorze graficznym w programie SEE Electrical Expert. Do badania modeli zastosowano środowisko Dedan (Deadlock analyzer) bazujące na logice temporalnej CTL (computational tree logic) i weryfikacji modelowej i umożliwiające identyfikację zakleszczeń całkowitych i częściowych oraz nieskutecznej terminacji agentów. Identyfikacja takich sytuacji pozwala określić błędy w zaprojektowanym urządzeniu lub jego powiązaniu.

Zostały sporządzone także modele elementów z uszkodzeniami: przerwą w cewce przekaźnika, spieczeniem zestyku.

Przeprowadzone na modelu eksperymenty potwierdziły możliwość zastosowania opracowanej metody do badania interfejsów i powiązań komputerowych urządzeń srk wykorzystujących przekaźniki jako elementy pośredniczące.

Słowa kluczowe: sterowanie ruchem kolejowym, weryfikacja temporalna, interfejs

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Summary

"The method for the preparation of interface specifications in the interconnection of computerized railway traffic control systems"

The thesis presents a method to prove the correctness of operation and cooperation of relay and computer railway traffic control devices, or to detect errors in the case of their incorrect design.

The logical layer of the interface of computer devices used to link these devices with other devices, such as relay devices, was considered. The method included the development of models of cooperating devices taking into account their characteristics relevant to the considered layer and potential faults in the elements of these devices.

The methodology presented in the paper successfully modeled relay railway traffic control systems and their interconnections with computer devices using the IMDS (Integrated Model of Distributed Systems) formalism. The models covered the basic elements for interconnections: contacts, coils, anchors, galvanic connections. The models were made in the form of graphs and code in the IMDS programming language. Bistable relays were included. The basic models were combined on the basis of connection statements achieved through the design of the circuit under modeling developed in the graphical editor in the SEE Electrical Expert program. The Dedan (Deadlock analyzer) environment, based on CTL (computational tree logic) temporal logic and model verification, was used to test the models and allows identification of total and partial deadlocks and ineffective termination of agents. Identification of such situations makes it possible to identify errors in the designed device or its interconnection.

Models of components with faults were also drawn up: relay coil break, contact baking.

Experiments carried out on the model confirmed the applicability of the developed method to the study of interfaces and interconnections of computerized railway traffic control devices using relays as intermediary elements.

Key words: railway traffic control, temporal verification, interface

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Spis treści

1 Wstęp ... 11
1.1 Wprowadzenie do zagadnienia ... 11
1.2 Pojęcia interfejsu i powiązania ... 11
1.3 Cele stosowania interfejsów i powiązań ... 14
1.4 Skala zagadnienia ... 15
1.5 Klasyfikacje powiązań ... 16
1.6 Model warstwowy interfejsu i powiązania ... 18
1.6.1 Warstwa fizyczna ... 20
1.6.2 Warstwa logiczna ... 23
1.6.3 Warstwa danych ... 27
1.7 Inicjatywy ujednolicenia interfejsów urządzeń srk ... 29
1.8 Metody badania układów przekaźnikowych ... 29
1.9 Cel i teza rozprawy ... 32
1.10 Zarys proponowanej metody ... 33
2 Weryfikacja formalna systemów ... 35
2.1 Wprowadzenie ... 35
2.2 Weryfikacja temporalna ... 37
2.3 Zintegrowany model systemów rozproszonych ... 39
2.4 Środowisko Dedan ... 44
2.5 Implementacja IMDS w języku programowania ... 45
2.6 Zakleszczenie i terminacja ... 46
2.7 Sprawdzanie modelu w środowisku Dedan ... 49
2.8 Notacja graficzna DA^3 ... 51
2.9 Weryfikacja w formalizmie IMDS ... 53
3 Uogólniony model interfejsu przekaźnikowego urządzeń komputerowych ... 56
3.1 Wprowadzenie ... 56
3.2 Wyjścia ... 56
3.3 Wejścia ... 66
4 Modele elementów składowych układu przekaźnikowego ... 69
4.1 Założenia ogólne ... 69
4.2 Model uzwojenia ... 71
4.3 Model kotwicy ... 73
4.4 Model zestyku ... 76
4.5 Założenia dla elementów z uszkodzeniami ... 78

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

4.6 Uszkodzenie uzwojenia przekaźnika ... 78
4.7 Spieczenie zestyku zwiernego przekaźnika ... 79
5 Specyfikacja układu przekaźnikowego w języku programowania ... 82
5.1 Zasady specyfikacji ... 82
5.1.1 Poziomy szczegółowości ... 82
5.1.2 Dołączanie serwerów do modelu ... 82
5.1.3 Nagłówek prototypu i typy danych ... 83
5.1.4 Zmienne ... 83
5.1.5 Wartości literalne ... 85
5.1.6 Odwołania do zmiennych ... 85
5.1.7 Funkcje ... 85
5.1.8 Inne podstawowe elementy języka opisu prototypu ... 86
5.2 Specyfikacja uzwojenia przekaźnika ... 86
5.2.1 Przykład wywołania ... 86
5.2.2 Nagłówek prototypu ... 87
5.2.3 Nagłówek serwera ... 87
5.2.4 Treść prototypu ... 87
5.3 Specyfikacja kotwicy przekaźnika ... 90
5.3.1 Przykład wywołania ... 90
5.3.2 Nagłówek prototypu ... 90
5.3.3 Nagłówek serwera ... 90
5.4 Specyfikacja zestyku ... 91
5.4.1 Przykład wywołania ... 91
5.4.2 Nagłówek prototypu ... 91
5.4.3 Nagłówek serwera ... 91
6 Przebieg badania powiązania ... 92
6.1 Założenia ... 92
6.2 Przykład powiązania ... 93
6.3 Opis układu pośredniczącego i wektorów sygnałów ... 95
6.4 Opis działania blokady ... 103
6.5 Specyfikacja interfejsu urządzeń komputerowych ... 109
6.5.1 Sygnały wejściowe do systemu komputerowego. ... 109
6.5.2 Sygnały wyjściowe z systemu komputerowego. ... 119
6.6 Badanie wybranego układu i jego powiązania ... 127
6.6.1 Wybór układu ... 127

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

6.6.2 Opracowanie schematu, zestawień i modelu ... 127
6.6.3 Określenie oczekiwanego działania ... 130
6.6.4 Uzupełnienie modelu o oczekiwane działanie ... 135
6.6.5 Wyniki eksperymentów ... 136
7 Omówienie rodzajów uszkodzeń elementów interfejsu ... 145
7.1 Wprowadzenie ... 145
7.2 Źródło klasyfikacji i rodzaje uszkodzeń ... 145
7.3 Omówienie podstawowych rodzajów uszkodzeń ... 150
8 Podsumowanie ... 152
8.1 Walidacja metody ... 152
8.1.1 Ręczna analiza pracy obwodu ... 152
8.1.2 Walidacja modeli ogólnych ... 152
8.2 Wnioski i kierunki dalszych prac ... 153
9 Spis tabel ... 156
10 Spis rysunków ... 157
11 Literatura ... 159

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

1 Wstęp

1.1 Wprowadzenie do zagadnienia

Sterowanie ruchem polega na regulowaniu ruchu pojazdów dla zapewnienia przewozów z efektywnym wykorzystaniem środków technicznych, przy zapewnieniu właściwego bezpieczeństwa i niezawodności procesu transportowego. Do środków technicznych w transporcie kolejowym można zaliczyć urządzenia sterowania ruchem kolejowym (srk). Powodowany postępem rozwój techniki obejmuje te urządzenia, wskutek czego klasyfikując je według sposobu realizacji zależności i sposobu przekazywania poleceń wyróżnić można urządzenia srk [19][69]:

- mechaniczne ręczne (in. kluczowe),
- mechaniczne pędniowe (in. scentralizowane)
- elektromechaniczne (in. suwakowe)
- elektryczne przekaźnikowe,
- elektryczne hybrydowe (przekaźnikowo-komputerowe),
- elektroniczne (elektryczne komputerowe).

Współcześnie wprowadzane nowe urządzenia srk są urządzeniami elektrycznymi w ujęciu powyższej klasyfikacji. Wprowadzane urządzenia obejmujące sobą pewien obszar wymagają co do zasady powiązania ich z urządzeniami istniejącymi wchodzącymi w skład otoczenia tego obszaru, w celu umożliwienia ich współpracy. Do zrealizowania takich powiązań wymagane są odpowiednie interfejsy łączonych urządzeń.

W pracy przedstawiono metodę pozwalającą wykazać poprawność działania i współdziałania przekaźnikowych i komputerowych urządzeń sterowania ruchem kolejowym, lub wykryć błędy w przypadku ich niepoprawnej konstrukcji.

1.2 Pojęcia interfejsu i powiązania

Interfejs jest pojęciem odnoszącym się do więcej niż jednego desygnatu. Słowo to jest spolszczeniem angielskiego słowa interface, które w języku polskim można przetłumaczyć bezpośrednio jako przejściówka lub sprzęg.

Dość ogólną definicję interfejsu odnaleźć można w encyklopedii [62]: interfejs, inaczej sprzęg lub złącze, jest pojęciem z dziedziny informatyki i dotyczy układu łączącego urządzenia systemu komputerowego w sposób umożliwiający ich współpracę. Może nim być również

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

program umożliwiający wymianę informacji pomiędzy programami. Definicja ta nie uwzględnia wszystkich potrzeb systemów automatyki kolejowej, gdzie często używa się jej do określenia wielu różnych miejsc wymiany informacji pomiędzy częściami systemu, na przykład blokadą liniową a urządzeniami stacyjnymi niekoniecznie wykonanymi w technice komputerowej [40].

W odniesieniu do urządzeń i systemów, w których występuje operator, w tym urządzeń i systemów kierowania i sterowania ruchem kolejowym (ksrk), pojęcie interfejsu wykorzystuje się również w odniesieniu do operatora. Wówczas oznacza ono charakterystyczny dla urządzenia zestaw urządzeń sterowniczych i sygnalizacyjnych: suwaków, pokręteł, przycisków, lampek, zegarów, monitorów umieszczonych na tablicy rozdzielczej, panelu, pulpicie itp. Interfejsy takie nazywane są często skrótem anglojęzycznym HMI (ang. human-machine interface – interfejs człowiek-maszyna) lub w odniesieniu do kierującego pojazdem kolejowym DMI (ang. driver-machine interface – interfejs kierowca-maszyna[1]).

Interfejs jest granicą urządzenia, w której lub przez którą dwie lub więcej części oddziałują na siebie [73]. Interfejs obejmuje również tę zaprojektowaną funkcję fragmentu urządzenia, która oddziałuje na, lub na którą oddziałuje zaprojekowana funkcja innego systemu [73]. Interfejs umożliwia łączenie i współpracę pomiędzy urządzeniami połączonymi za jego pośrednictwem. Współpraca rozumiana jest jako przenoszenie energii i wymiana sygnałów o określonej postaci. Interfejs może opierać się o współpracę na drodze mechanicznej, elektrycznej, elektronicznej lub optycznej.

Termin ten powinien odnosić się do informatyki, czyli do przekazywania sygnałów. Definicja encyklopedyczna dotyczy jednak układu komputerowego, co w odniesieniu do dziedziny sterowania ruchem kolejowym jest niewystarczające. Urządzenia sterowania ruchem kolejowym wykonywane były i są w taki sposób, że do przesyłania i przetwarzania sygnałów wykorzystują nie tylko sprzęt komputerowy, ale ogólniej elektryczny (na przykład przekaźniki), mechaniczny lub inny.

Cechy, które należy uwzględnić w definicji, są następujące:

[1] Tłumaczenie własne. Słowo driver powinno zostać przetłumaczone w zależności od sytuacji w różny sposób. Kierowca jest określeniem właściwszym dla transportu drogowego. W ruchu kolejowym, zgodnie z powszechnie stosowaną terminologią, pojazdem kolejowym kieruje maszynista (dawniej nazywany również mechanikiem).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- Konstrukcja może wykorzystywać różną technologię – niekoniecznie komputerową, ale na przykład mechaniczną, elektryczną przekaźnikową, optyczną i tym podobne.
- Pojęcie obejmuje swoim zakresem układ wejść i wyjść – w danym przypadku może obejmować i wejścia i wyjścia albo wyłącznie wejścia lub wyłącznie wyjścia.
- Źródła przekazywanej informacji nie są wyłącznie fizycznymi wejściami i wyjściami – potrzeba także wyodrębnienia logiki działania źródeł uwzględnionej w logice urządzenia, dla którego definiuje się interfejs, czyli opisu sygnałów fizycznych i odpowiadającego im algorytmu działania układu wejść i wyjść w kontekście algorytmów pracy urządzenia i innego urządzenia mającego (innych urządzeń mających) współpracować poprzez ten interfejs z danym urządzeniem.

Należy również zwrócić uwagę, że powyższe cechy dotyczą elementu (często złożonego) danego urządzenia. W sytuacji, kiedy łączone są dwa urządzenia przy pomocy ich interfejsów powstaje nowa struktura – układ powiązania, którego elementami są interfejsy zaangażowanych urządzeń wraz z zasadami ich interakcji.

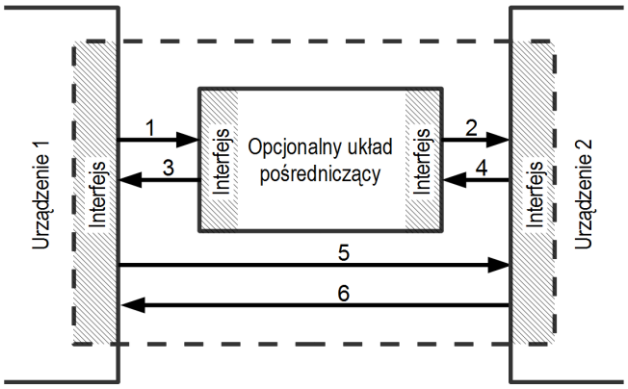
Biorąc pod uwagę powyższe w odniesieniu do projektów srk, w tym tych obejmujących instalacje ETCS (ang. European Train Control System – Europejski System Sterowania Pociągiem), autorzy artykułu [40] proponują rozróżnienie pojęć interfejsu i powiązania, i ich następujące definicje.

Interfejs – jest to układ wejść i wyjść danego urządzenia (systemu, podsystemu) wraz z sygnałami przez nie przesyłanymi oraz odpowiadającą im logiką i sekwencjami działania urządzenia, umożliwiający łączenie i współpracę pomiędzy tym urządzeniem a innymi urządzeniami.

Powiązanie – jest to układ połączeń wykorzystujący interfejsy urządzeń, mogący zawierać dodatkowe elementy, który umożliwia łączenie i współpracę pomiędzy urządzeniami połączonymi za jego pośrednictwem. Współpraca rozumiana jest jako przenoszenie energii i/lub wymiana sygnałów o określonej postaci.

Powiązanie jest zatem pojęciem szerszym, obejmującym zawsze interfejsy łączonych urządzeń. Interfejs natomiast nie jest czymś uniwersalnym, ponieważ zaprojektowano go z myślą o wykorzystaniu do powiązania z konkretnym urządzeniem lub przekazywania określonych sygnałów do wykorzystania przy powiązaniu z określoną klasą urządzeń. Określony rodzaj interfejsu można wykorzystywać zatem do różnych powiązań. Rozróżnienie pojęć zilustrowano rysunkiem 1.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21



Rysunek 1. Idea współpracy urządzeń srk. 1-6 - możliwe interakcje. Źródło: [40]

1.3 Cele stosowania interfejsów i powiązań

Celem stosowania interfejsu jest konieczność zapewnienia możliwości i miejsca współpracy danego urządzenia, w skład którego wchodzi interfejs, z otoczeniem. Brak interfejsu uniemożliwiałby przekazywanie sygnałów pomiędzy urządzeniem a otoczeniem, co utrudniałoby lub uniemożliwiało pracę tego urządzenia.

Powody niekompatybilności współpracujących urządzeń mogą być różne. Jako cele stosowania powiązań można wskazać:

- uproszczenie konstrukcji urządzenia posiadającego opracowany interfejs,
- potrzebę separacji urządzeń np. separację galwaniczną,
- zapewnienie wygody i szybkości obsługi.

Za pierwszym powodem mogą się kryć nieuwzględnienie przez konstruktora pewnych innych lub nowszych rozwiązań technicznych lub celowe ich pominięcie ze względów praktycznych lub ekonomicznych. Nie opracowuje się doskonale uniwersalnych rozwiązań.

Separacja urządzeń może być związana z potrzebą podziału urządzeń na moduły, co jest z kolei związane z względami łatwości ich wymiany podczas naprawy. Innym powodem może być potrzeba zapewnienia nie przenoszenia się pomiędzy urządzeniami usterek związanych z zakłóceniami sygnałów. Może być to podyktowane zarówno względami bezpieczeństwa pracy urządzeń i osób je obsługujących lub użytkujących, jak również niezawodności systemu, w skład którego wchodzą (nie przenoszenie się określonych usterek z urządzenia do urządzenia).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Ostatni powód związany jest z interfejsami operatora. Przykładem może być odczyt wskazania zegara – łatwiejszy, gdy posiada on cyferblat niż gdyby posiadał diody wskazujące binarnie godzinę.

1.4 Skala zagadnienia

Interfejs można zdefiniować dla różnej skali:

- małej – np. pomiędzy napędem a nastawnicą (wówczas będzie nim układ nastawczo-kontrolny),
- dużej – np. pomiędzy dwoma centrami zdalnego sterowania.

Pośrednie stopnie możliwe są również do określenia.

Podobnie można interfejsem nazwać jedynie łącze składające się z przewodów, gniazd i wtyczek, zacisków umieszczonych w drodze sygnału elektrycznego. Traktując zagadnienie coraz szerzej należałoby rozważyć elementy dostosowujące sygnały do tego łącza: elementy elektryczne, optoelektroniczne, dodatkowe przekaźniki zastosowane do obsługi interfejsu w systemie, wreszcie części oprogramowania służące obsłudze interfejsu.

Systemy sterowania ruchem kolejowym dzieli się zgodnie z przeznaczeniem funkcjonalnym na [19]:

- stacyjne
- liniowe:
    - blokady liniowe,
    - na przejazdach kolejowo-drogowych (zabezpieczające ruch pojazdów na jednopoziomowych skrzyżowaniach dróg kołowych i torów),
    - przekazywania informacji w relacji tor-pojazd (bezpiecznej kontroli jazdy pociągu),
    - specjalne.

Wobec tego opracowując system będący jednym z powyższych projektanci stają przed zagadnieniem odpowiedniego ukształtowania wejść i wyjść tego systemu umożliwiających współpracę z innymi wyżej wymienionymi systemami. Choć interfejs można określić, jak podano powyżej, dla różnej skali zagadnienia, to najczęściej wykorzystywanym przez projektantów branży srk desygnatem tego terminu jest właśnie układ wejść i wyjść systemu tak szerokiego jak np. stacyjne urządzenia srk lub blokada liniowa.

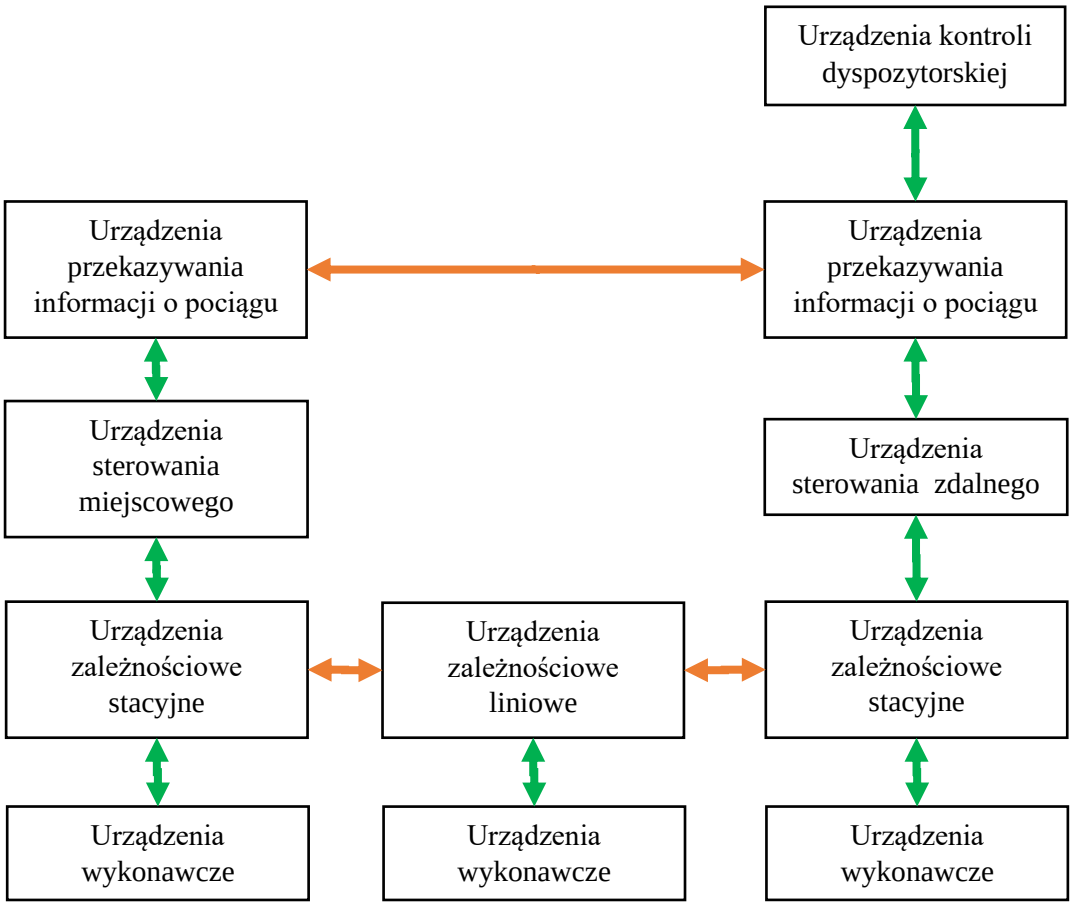
Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

W pracy pojęcie interfejsu zostanie celowo zawężone do interfejsów zewnętrznych systemów srk tak rozległych i złożonych, jak systemy stacyjne, blokady liniowe, urządzenia samoczynnej sygnalizacji przejazdowej. Praca nie będzie dotyczyć interfejsów przewidzianych do współpracy systemów z operatorami ani interfejsów w niższych szczeblach struktury hierarchicznej elementów, układów i systemów ksrk. W dalszym tekście, o ile nie wyszczególniono inaczej, tak właśnie ograniczone zostanie pojęcie interfejsu.

1.5 Klasyfikacje powiązań

Według relacji w warstwowym modelu ksrk [77] wyróżnić można powiązania (rysunek 2):

- pionowe,
- poziome.



Rysunek 2. Powiązania poziome (pomarańczowe) i pionowe (zielone) urządzeń ksrk w warstwowym modelu ksrk. Źródło: opracowanie własne na podstawie [77]

Według poziomu bezpieczeństwa wymienianych sygnałów wyróżnić można (na podstawie faktu istnienia sygnałów o takim charakterze) [66]:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- powiązania zwykłe (tzw. niekrytyczne, informacyjne),
- powiązania bezpieczne (tzw. krytyczne, zależnościowe).

Definicja interfejsu sformułowana w rozdziale 1.2 nie nakłada ograniczeń, na to, w jakiej technologii ma być wykonany interfejs. W praktyce jednak, w obecnie wdrażanych systemach sterowania ruchem kolejowym stosuje się interfejsy wykorzystujące elektryczne sygnały cyfrowe.

W przypadku współpracy urządzeń komputerowych z komputerowymi lub przekaźnikowymi możliwe są różne realizacje interfejsów i zbudowanych w oparciu o nie powiązań. Wskazać można powiązania [77]:

- przekaźnikowe:
    - poleceniowe przekaźnikowo-przekaźnikowe (obydwa interfejsy są przekaźnikowe), dla których:
        - sterowanie elementem wykonawczym (cewka po stronie urządzeń odbierających polecenie) odbywa się bezpośrednio przez zestyki,
        - sterowanie elementem wykonawczym (cewka) odbywa się przez zestyki i elementy adresowe (zestyki elektroniczne wykonane w postaci tyrystorów albo triaków),
    - poleceniowe przekaźnikowo-elektroniczne (interfejsy: przekaźnikowy – cewka po stronie urządzeń odbierających, i elektroniczny – np. transoptor po stronie urządzeń nadających polecenie),
    - meldunkowe (interfejsy: elektroniczny po stronie urządzeń odbierających polecenie w postaci np. transoptora, i przekaźnikowy po stronie urządzeń nadających polecenie w postaci zestyku),
- elektroniczne:
    - komunikacyjne (inaczej: szeregowe), w których przekazywanie komunikatów wymaga ustalonego protokołu cyfrowej wymiany danych,
    - binarne (inaczej: równoległe), w których brak możliwości elektronicznej wymiany danych wymaga elementów stykowych w interfejsach, a współpraca z urządzeniami opartymi o maszyny stanów wymaga specjalnego interfejsu (np. zestawu przekaźników).

Przykładami powiązań przekaźnikowych są powiązania urządzeń komputerowych stacyjnych z popularnymi na sieci zarządzanej przez PKP PLK S.A. (PKP Polskie Linie Kolejowe S.A.)

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

przekaźnikowymi blokadami liniowymi takimi jak blokady typu[2] Eac i Eap (interfejsy poleceniowe przekaźnikowo-elektroniczne i meldunkowe), oraz urządzeniami zabezpieczającymi ruch na jednopoziomowych skrzyżowaniach kolejowo-drogowych (różne powiązania przekaźnikowe). Komputerowe blokady liniowe (np. typu SHL produkcji firmy Bombardier, CBL-2000 produkcji firmy Rail-Mil Computers) współpracują również z przekaźnikowymi i mechanicznymi urządzeniami stacyjnymi za pośrednictwem powiązań przekaźnikowych. Przykładami współpracy w oparciu o powiązanie elektroniczne może być współpraca blokady liniowej CBL-2000 i urządzeń MOR-3 (komputerowy system stacyjnych urządzeń zależnościowych produkcji firmy Kombud).

Powiązania urządzeń komputerowych z przekaźnikowymi charakteryzują następujące fakty [77]:

- wymagają znajomości systemu przekaźnikowego i komputerowego,
- wymagają stosowania elementów pośredniczących (przekaźniki pośredniczące, płytki przejściowe),
- często należy zaprojektować powtarzacze (z powodu braku wymaganej liczby i/lub rodzaju wolnych zestyków przekaźników systemu przekaźnikowego niezbędnych do wykonania powiązania),
- są stosunkowo czytelne dla projektantów i monterów,
- pozwalają przekazywać tylko stany binarne,
- są ograniczone odległością pomiędzy urządzeniami (systemami).

1.6 Model warstwowy interfejsu i powiązania

Większość sieci jest zorganizowana w postaci stosu warstw (inaczej poziomów), z których każda jest zbudowana ponad poprzednią. Zadaniem każdej warstwy jest oferowanie określonych usług warstwom wyższym i izolowanie ich od szczegółów faktycznej implementacji świadczonych usług. Określona warstwa w jednej maszynie prowadzi konwersację z tożsamą warstwą w innej maszynie według określonych reguł i konwencji. Zbiorami reguł ustalających znaczenie i format pakietów oraz wiadomości wymienianych pomiędzy równorzędnymi jednostkami w obrębie warstwy są protokoły. Jednostki te używają

[2] Typ urządzenia to, zgodnie z definicją zawartą w ustawie o transporcie kolejowym [70], urządzenie lub system przeznaczone do prowadzenia ruchu kolejowego o określonych powtarzalnych parametrach technicznych i eksploatacyjnych. Podany typ charakteryzuje konkretne urządzenie i często jest nazwą literowo-cyfrową, jak w podanych w akapicie przykładach (Eac, Eap, SHL, CBL-2000, MOR-3).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

protokołów do implementacji definicji swoich usług świadczonych na rzecz warstwy wyższej [68].

Powszechnie stosowany jest warstwowy model odniesienia OSI (pełna nazwa ISO OSI RM, ang. ISO Open Systems Interconnection Reference Model – model odniesienia łączenia systemów otwartych; ISO – ang. International Organization for Standardization – Międzynarodowa Organizacja Normalizacyjna) zawierający 7 warstw oraz model TCP/IP (ang. Transmission Control Protocol/Internet Protocol) zawierający 4 warstwy.

Przy opisie interfejsu i powiązania można wyróżnić następujące warstwy [36]:

- fizyczną,
- logiczną,
- danych.

Podział na warstwy jest naturalnym sposobem reprezentacji struktury urządzenia i zasad jego funkcjonowania. Pozwala na szczegółowe i uporządkowane przedstawienie interfejsów powiązania a następnie analizę powiązania zarówno w odniesieniu do interesującej warstwy, jak i z punktu widzenia interakcji pomiędzy cechami w poszczególnych warstwach.

Warstwa fizyczna jest warstwą opisu zawierającą dane na temat właściwości fizycznych - elektrycznych i mechanicznych - interfejsu i powiązania.

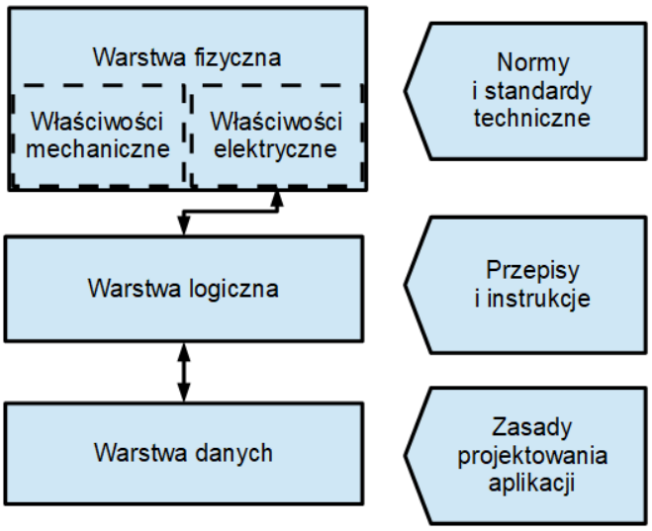
Warstwa logiczna zawiera wykaz sygnałów z ich opisem (znaczenie sygnałów). Określonym sygnałom wykorzystywanym w warstwie logicznej są przyporządkowane cechy wynikające z warstwy fizycznej. Ponadto wykorzystywana jest reprezentacja w postaci automatu. Automat ten reprezentuje urządzenie i jego zachowanie obserwowane od strony interfejsu.

Warstwa danych zawiera opis zachowania urządzenia widziany poprzez interfejs. Dogodnym sposobem jej opisu może być, podobnie jak w warstwie logicznej, reprezentacja w postaci automatu albo diagram sekwencji opracowany w języku SysML (ang. The Systems Modeling Language).

Podział na trzy warstwy może zostać rozwinięty o kolejne w miarę wzrostu skomplikowania analizowanego powiązania.

Warstwy powiązania przedstawiono graficznie na rysunku 3 i omówiono szerzej w kolejnych rozdziałach.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21



Rysunek 3. Warstwy powiązania i wykorzystywane do ich określenia zbiory dokumentów. Źródło: opracowanie własne.

1.6.1 Warstwa fizyczna

Jak zaznaczono w poprzednim rozdziale warstwa fizyczna zawiera opis właściwości fizycznych. Właściwości te można podzielić na:

- elektryczne,
- mechaniczne.

Warto zauważyć przy tym, że powszechne zastosowanie mają sygnały elektryczne przekazywane poprzez połączenia galwaniczne. Inne, fizycznie znane sposoby przekazywania informacji, też byłyby możliwe, ale nie spotyka się ich powszechnie (przekazywanie falami, w szczególności radiowymi, ruchem elementów mechanicznych, przekazywanie połączeniami optycznymi takimi jak światłowody, czy elementy transoptorów itd.). Stąd w pracy nacisk jest położony właśnie na połączenie elektryczne.

Właściwości elektryczne połączenia wymagają określenia dopuszczalnych zakresów parametrów podstawowych takich jak:

- napięcie i prąd oraz ich wartości maksymalne,
- rodzaj sygnału elektrycznego (stały, zmienny, przemienny) wraz z określeniem cech charakteryzujących go jak częstotliwość, przyjęty rodzaj modulacji itp.

Sygnały elektryczne powinny zostać dobrane m.in. z uwzględnieniem potencjalnych zakłóceń, a zatem zależeć będą od właściwości otoczenia powiązania. Pewne zakłócenia mogą zostać wyeliminowane lub stłumione, dzięki odpowiedniemu doborowi właściwości elektrycznych i fizycznych powiązania. Inne wymagają dodatkowych zabezpieczeń (ekranów, modulacji, środków ochrony przeciwprzepięciowej i przeciw przetężeniowej).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Właściwości mechaniczne połączenia wymagają określenia rodzaju gniazda, wtyczki, zacisku lub innego elementu mechanicznego umożliwiającego połączenie. Określają również cechy materiałów zastosowanych jako przewodniki i izolatory. Na właściwości mechaniczne wpływ będzie miało to, na jakie narażenia mechaniczne, chemiczne, biologiczne ma być odporne powiązanie. Zastosowane materiały muszą również spełniać szereg innych wymagań związanych z ich uciążliwością dla środowiska naturalnego na etapie produkcji, eksploatacji i utylizacji. Istotne znaczenie ma przewidywanie łatwości rozłączania i łączenia połączenia przy użyciu określonych narzędzi (cechy związane z ergonomią) oraz niezawodności połączenia w funkcji czasu i liczbie cykli połącz/rozłącz dla połączeń gniazdo-wtyk (trwałość).

Warto zauważyć, że właściwości mechaniczne i elektryczne są od siebie w pewnych zakresach zależne. Dla przykładu maksymalne natężenie prądu będzie wpływać na przekrój przewodnika stosowanego do wykonania połączenia, z kolei maksymalna wartość napięcia będzie wpływać na grubości i materiał zastosowanych izolatorów lub odstępy powietrzne i powierzchniowe pomiędzy elementami o różnym potencjale [57]. Brak fizycznej możliwości zastosowania określonego materiału lub wymiarów połączenia może wobec tego wpływać na ograniczenia, jakie zostaną nałożone na wartości sygnałów elektrycznych.

Przykładem istotności omawianych cech może być system WT UZ [66] (elektroniczne urządzenia nastawcze z komputerowymi urządzeniami zależnościowymi typu WT UZ), gdzie zastosowano standardowe kasety komputerów przemysłowych. Kasety te zawierają karty na tyle wąskie, że zmieszczono na nich gniazda D-SUB50 (ang. D-subminiature). Gniazda te nie nadają się do bezpośredniego podłączenia przewodów (standardowe żyły o przekroju $1mm^2$). Konieczne stało się zastosowanie tzw. płytek przejściowych – płytek drukowanych zawierających z jednej strony wtyczkę D-SUB a na powierzchni wlutowanych po obwodzie zacisków umożliwiających podłączenie pojedynczych żył o różnym przekroju, w tym $1mm^2$.

W przypadku powiązania koderów LEU (ang. Lineside Electronic Unit – koder dla Eurobalis) z urządzeniami warstwy podstawowej (poprzez interfejs S kodera do obwodów nastawczo-kontrolnych sygnalizatorów przytorowych) warstwa fizyczna składa się z zacisków elektrycznych, do których należy podłączyć przewody idące do sygnalizatora przytorowego i układu nastawczo-kontrolnego tego sygnalizatora. Wewnątrz kodera zaciski są połączone parami poprzez elementy elektryczne umożliwiające pomiar wartości elektrycznych w obwodach (rezystor pomiarowy, przekładnik prądowy lub inny układ pomiarowy taki, jak np. czujnik Halla).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Powyższe rozważania można przedstawić w postaci następującej listy parametrów:

- parametry mechaniczne:
    - liczba wyprowadzeń umożliwiających podłączenie przewodów elektrycznych przypadających na dane wejście lub wyjście,
    - liczba obsługiwanych wejść i wyjść,
    - grubość żyły,
    - grubość izolacji żyły,
    - liczba kabli wprowadzonych do obudowy,
    - grubość kabla,
    - możliwość podłączenia ekranu kabla,
    - długość kabla potrzebna do rozszycia mierzona do wewnątrz od miejsca wprowadzenia do obudowy,
- parametry elektryczne:
    - maksymalne natężenie prądu w obwodzie wejścia lub wyjścia (wartość skuteczna),
    - maksymalny spadek napięcia na wejściu lub wyjściu (wartość skuteczna),
    - dopuszczalne zakłócenia sygnałów (wynikające z rozwiązania technicznego włączania i wyłączania napięcia lub prądu w obwodzie wejścia lub wyjścia) – wyrażone w postaci ich opisu lub wskazania tolerowanego czasu trwania części sygnału nie traktowanego jako zakłócenie,
    - zakres mierzonego kąta przesunięcia fazowego pomiędzy napięciem i natężeniem w obwodzie odbiornika,
    - impedancja (lub rezystancja) żył względem siebie i otoczenia (np. ziemi, w której może być ułożony kabel powiązania).

Powyższe parametry dotyczą interfejsów wszystkich urządzeń integrowanych w powiązaniu. Określenie tych parametrów warstwy fizycznej wynika z natury sygnałów elektrycznych i obwodów służących do ich przesyłania. Aby energia sygnału elektrycznego mogła być wykorzystana w odbiorniku, wspomniane impedancje żył obwodu względem siebie i ziemi muszą być dostatecznie duże. Zbyt małe wartości impedancji powodować będą wzrost natężenia prądu płynącego pomiędzy żyłami i pomiędzy żyłami i ziemią a wraz z tym wzrost strat energii. Zbyt mała wartość impedancji obwodu będzie powodować także nadmierny wzrost natężenia prądu i, po przekroczeniu określonej wartości, zadziałanie bezpiecznika przerywającego obwód.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Wartości parametrów elektrycznych mogą zostać wyrażone w postaci zbioru nieprzeliczalnego o określonych granicach przedstawione dla przykładu interfejsu S kodera LEU w postaci tabeli (tabela 1 kolumny 1.-5.).

1.6.2 Warstwa logiczna

Warstwa logiczna zawiera opis sygnałów i ich przeznaczenie. Przyporządkowuje również sygnałom właściwości określone w warstwie fizycznej.

Możliwe jest przechowywanie informacji o wzorcu sygnału wejściowego ale częstszym rozwiązaniem jest takie skonstruowanie lub zaprogramowanie urządzenia, aby ignorowało zmiany sygnałów wejściowych odbiegających od wzorca utrzymujące się w pewnym dość krótkim czasie. Określonym wartościom sygnałów elektrycznych przyporządkowane zostają stany logiczne (binarne). Przykład dla interfejsu S kodera LEU przedstawiono w tabeli 1.

Tabela 1. Przykładowe przyporządkowanie określonym wartościom sygnałów elektrycznych sygnałów binarnych dla interfejsu S kodera LEU. Źródło: [36]

Lp.	Parametr	Miano	Wartość		Wartość logiczna
			minimalna	maksymalna	
1.	*2.*	*3.*	*4.*	*5.*	*6.*
Komora z pojedynczą żarówką					
1.	Napięcie	V	0	15	0
2.			110	135	1
3.			Pozostałe zakresy		Wejście uszkodzone
4.	Natężenie	mA	0	1	0
5.			200	220	1
6.			Pozostałe zakresy		Wejście uszkodzone
7.	Częstotliwość	Hz	0	45	0
8.			45	65	1
9.			Pozostałe zakresy		Wejście uszkodzone
Komora z żarówką główną i rezerwową z opornikiem					
…	…	…	…	…	…

Powyższe przyporządkowanie można przedstawić także w postaci zależności (1), (2) i (3).

$$we_1^{log} = \text{"1"} \Leftrightarrow U_1 \in \langle 110V; 135V\rangle \wedge I_1 \in \langle 200mA; 220mA\rangle \wedge f_1 \in \langle 45Hz; 65Hz\rangle \quad (1)$$

$$we_1^{log} = \text{"0"} \Leftrightarrow U_1 \in \langle 0V; 15V\rangle \wedge I_1 \in \langle 0mA; 1mA\rangle \wedge f_1 \in \langle 0Hz; 45Hz\rangle \quad (2)$$

$$we_1^{log} = \text{"}wejście\ uszkodzone\text{"} \Leftrightarrow U_1 \notin (\langle 0V; 15V\rangle \cup \langle 110V; 135V\rangle) \vee I_1 \notin (\langle 0mA; 1mA\rangle \cup \langle 200mA; 220mA\rangle) \vee f_1 \notin \langle 0Hz; 65Hz\rangle \quad (3)$$

gdzie:

we_1^{log} – stan logiczny wejścia 1,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

U_1 – wartość skuteczna napięcia na wejściu 1,

I_1 – wartość skuteczna natężenia prądu na wejściu 1,

f_1 – częstotliwość na wejściu 1.

W przypadku kodera LEU sygnałom binarnym w określonej kombinacji odpowiada wykrycie określonego wskazania sygnalizatora. Przyporządkowanie wskazań sygnałom binarnym wykrytym na określonych wejściach wymaga jeszcze przyporządkowania stanów logicznych komór spośród następujących:

- ON – komora świeci światłem ciągłym, wówczas gdy stan logiczny wynosi „1”,
- OFF – komora nie świeci, wówczas gdy stan logiczny wynosi „0”
- FLASH – komora wyświetla sygnał migający, wówczas gdy stan logiczny wynosi naprzemiennie „0” i „1”,
- USZKODZONA – komora uszkodzona, wówczas gdy stan logiczny wynosi „wejście uszkodzone”.

Do określenia prawidłowości migania światła (przemienne sygnały binarne) konieczne jest również określenie dopuszczalnej częstotliwości migania i wypełnienia impulsu. Wartości te są określone np. w instrukcji Ie-12 [30] („*częstotliwość świecenia światła migającego powinna wynosić w granicach od 0,8Hz do 1,25Hz, co odpowiada 48 do 75 włączeń na minutę. Współczynnik wypełnienia określony jako stosunek czasu świecenia światła do czasu przerwy w świeceniu powinien wynosić od 2:3 do 3:2*.”). Stąd:

- Częstotliwość migania komory $f_{mig} \in \langle 0{,}8Hz; 1{,}25Hz \rangle$,
- Współczynnik wypełnienia $k_w \in \langle \frac{2}{3}; \frac{3}{2} \rangle$ (w dalszym wzorze dla uproszczenia przyjęto 1/1).

Na tej podstawie można określić w postaci zależności (4), (5) i (6) przyporządkowanie stanów logicznych stanom komór.

$$we_n = \text{"ON"} \Leftrightarrow x_n(t) = 1 \wedge (x_n(t) = 1) \triangleq \left(we_n^{log} = \text{"1"}\right) \quad (4)$$

$$we_n = \text{"OFF"} \Leftrightarrow x_n(t) = 0 \wedge (x_n(t) = 0) \triangleq \left(we_n^{log} = \text{"0"}\right) \quad (5)$$

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

$$we_n = \text{"FLASH"} \Leftrightarrow x_n(t) = \frac{sgn(\sin 2\Pi ft) + 1}{2} \wedge f \in \langle 0{,}8Hz; 1{,}25Hz\rangle \wedge (x_n(t) = 1) \triangleq \left(we_n^{log} = \text{"1"}\right) \wedge (x_n(t) = 0) \triangleq \left(we_n^{log} = \text{"0"}\right) \quad (6)$$

Gdzie:

we_n - stan n-tej komory,

$x_n(t)$ - funkcja czasu sygnału logicznego *n*-tego wejścia.

Przyporządkowanie wskazań sygnalizatora do kombinacji stanów komór może zostać przedstawione w postaci tabeli prawdy opracowanej na podstawie dokumentów formalnych opisujących wskazania sygnalizatorów [65][32]. Przykładowa tabela dla kodera LEU została przedstawiona w tabeli 2.

Tabela 2. Przykładowa tabela prawdy dla wskazań sygnalizatora zawierająca w komórkach stany komór we_n. Źródło: opracowane własne na podstawie [36]

Numer komory n	Opis komory	Wskazanie S sygnalizatora				
		S1	S2	…	S7	…
1.	*2.*	*3.*				
1	Komora światła czerwonego	ON	OFF	…	OFF	…
2	Komora światła zielonego	OFF	ON	…	FLASH	…
3	Komora światła pomarańczowego górnego	OFF	OFF	…	OFF	…
4	Komora światła pomarańczowego dolnego	OFF	OFF	…	ON	…
5	Komora światła białego	OFF	OFF	…	OFF	…
6	Komora pasa zielonego	OFF	OFF	…	ON	…
…	…	…	…	…	…	…

Powyższe przyporządkowanie można przedstawić także w postaci zależności (7).

$$S = \begin{cases} S1 \Leftrightarrow we_1 = \text{"ON"} \wedge we_2 = \text{"OFF"} \wedge we_3 = \text{"OFF"} \wedge we_4 = \text{"OFF"} \wedge we_5 = \text{"OFF"} \wedge we_6 = \text{"OFF"} \wedge \ldots \\ S2 \Leftrightarrow we_1 = \text{"OFF"} \wedge we_2 = \text{"ON"} \wedge we_3 = \text{"OFF"} \wedge we_4 = \text{"OFF"} \wedge we_5 = \text{"OFF"} \wedge we_6 = \text{"OFF"} \wedge \ldots \\ \ldots \\ S7 \Leftrightarrow we_1 = \text{"OFF"} \wedge we_2 = \text{"FLASH"} \wedge we_3 = \text{"OFF"} \wedge we_4 = \text{"ON"} \wedge we_5 = \text{"OFF"} \wedge we_6 = \text{"ON"} \wedge \ldots \\ \ldots \end{cases} \quad (7)$$

gdzie:

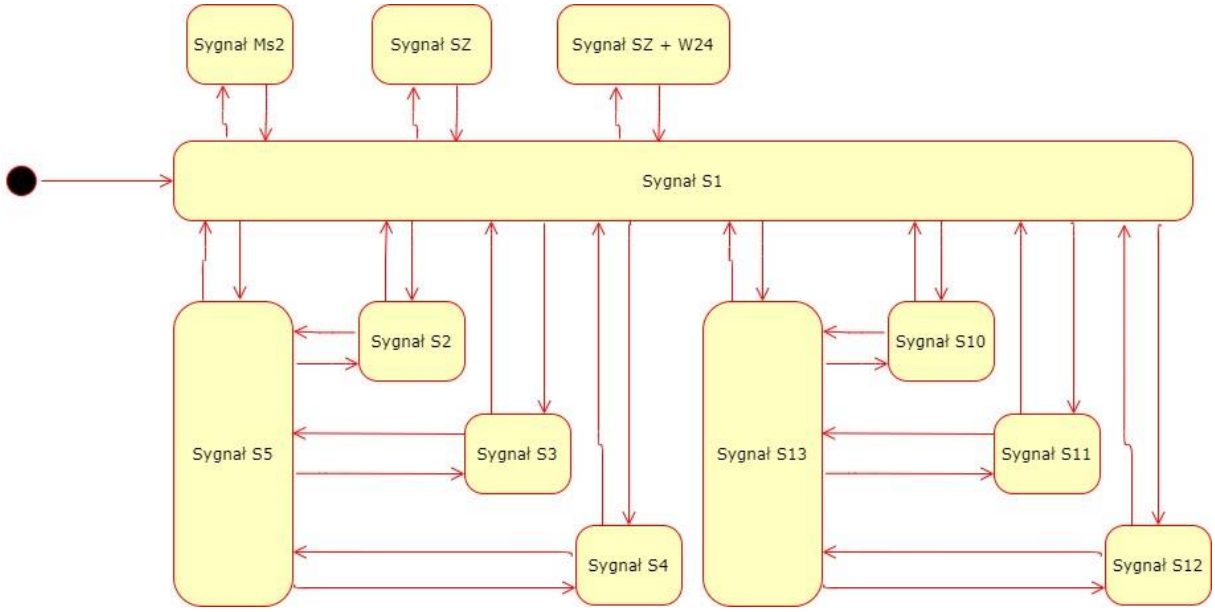
S - wskazanie sygnalizatora,

$S1$, $S2$, … - możliwe obrazy sygnałowe.

Warstwa logiczna zawiera również (o ile urządzenie opisywane poprzez interfejs działa w taki sposób) opis dopuszczalnych przejść pomiędzy stanami np. w postaci grafu lub tabeli przejść.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Jest to reprezentacja automatu Moore'a. Przykład grafu dla kodera LEU przedstawiono na rysunku 4.



Rysunek 4. Przykładowy graf reprezentujący możliwe zmiany stanów sygnałów odczytywanych poprzez koder LEU. Źródło [36]

Przedstawione na rysunku 4 stany i możliwe przejścia pomiędzy nimi reprezentują działanie przykładowych urządzeń na stacji sterujących wskazaniami jednego sygnalizatora. Czarną kropką oznaczono stan początkowy, występujący po uruchomieniu urządzeń. Pozostałe stany odpowiadają możliwym obrazom sygnałowym. Warto zwrócić uwagę, że graf ten może reprezentować działanie interfejsu urządzeń warstwy podstawowej i oczekiwane przez koder działanie tych urządzeń. Koder, ze względu na sposób w jaki przyporządkowuje sygnałom telegramy, co opisano w kolejnym rozdziale, może nie analizować prawidłowości przechodzenia pomiędzy stanami sygnalizatora. Jest to rozwiązanie stosowane również w innych powiązaniach (np. komputerowych urządzeń zależnościowych typu WT UZ z samoczynną przekaźnikową blokadą liniową typu Eac).

Graf przedstawiony na rysunku 4 nadaje się również do opisu interfejsu urządzeń srk warstwy podstawowej (urządzenia współpracujące poprzez powiązanie z koderem LEU). Przejścia pomiędzy stanami powinny zostać uzupełnione o warunki wynikające z działania układu nastawczo-kontrolnego (polecenia nastawcze oraz reakcja na zmiany parametrów elektrycznych takich jak napięcie, natężenie i impedancja elementów obwodu kontrolowanych przez układ). Przykładowo wykrycie natężenia prądu poniżej zdefiniowanego zakresu może oznaczać przerwę w obwodzie lub usterkę żarówki. Wykrycie natężenia prądu powyżej zdefiniowanego zakresu może oznaczać zwarcie w obwodzie. W przypadku sygnału S12

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

spowoduje to nastawienie sygnału S1 (tj. reakcja dotycząca przejścia pomiędzy stanami S12 i S1 na rysunku 4).

1.6.3 Warstwa danych

Warstwa danych jest najwyższą ze zdefiniowanych warstw. Sformułowane są w niej na zwiększonym poziomie ogólności dane, które mają trafiać do interfejsu lub pochodzić od niego oraz odpowiadające im reakcje urządzenia współpracującego.

Obwód nastawczo-kontrolny sygnalizatora służy do nastawienia właściwego sygnału na sygnalizatorze po spełnieniu odpowiednich warunków zależnościowych i ewentualnie wydaniu poleceń przez personel. Sygnał ten odnosi się do warunków jazdy i zależy od tego czy jazda ma być pociągową, czy manewrową, oraz od rodzaju sygnalizatora. Można przeprowadzić ogólny podział na:

- sygnały zezwalające na jazdę,
- sygnały zabraniające jazdy,
- sygnały uprzedzające o powyższych sygnałach.

Szczegółowe znaczenie sygnałów jest opisane na szczeblu krajowym w rozporządzeniu [65] i doprecyzowane w instrukcjach zarządców infrastruktury (np. [32]). Opis ten jest opisem słownym wyjaśniającym możliwości wykonywania ruchu na podstawie określonego wskazania sygnalizatora. Wyróżniono przy tym również tzw. sygnał wątpliwy, czyli taki, który nie odpowiada swoim obrazem pozostałym opisanym sygnałom. Sygnał taki odbiorca (w szczególności kierujący pojazdem maszynista) ma obowiązek traktować jako sygnał zabraniający jazdy.

Układ nastawczo-kontrolny umożliwia również kontrolę prawidłowości wskazań sygnalizatora w odniesieniu do wymuszenia nastawienia określonego sygnału zadanego przez układ nastawczo-kontrolny i współpracujące z nim układy. W przypadku niezgodności uzyskanych wskazań względem wymuszonych wynikającej z możliwych usterek (np. przepalenia się żarówki w komorze) układy te mają za zadanie nastawić sygnał bezpieczniejszy (ograniczający prędkość), który w szczególności może być sygnałem zabraniającym jazdy.

Oprócz zgodności sygnału z obrazem przedstawionym w przytoczonych dokumentach formalnych musi istnieć zgodność z warunkami wykonywania ruchu. Dlatego też każdy sygnalizator ma na etapie projektowania aplikacji urządzeń srk przyporządkowany ograniczony

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

zbiór wskazań dopuszczalnych będący podzbiorem zbioru wszystkich możliwych wskazań, co można przedstawić w postaci zależności (8).

$$\boldsymbol{S}_e \subset \boldsymbol{S} \tag{8}$$

Gdzie:

$\boldsymbol{S}$ – zbiór wszystkich dopuszczalnych wskazań sygnalizatorów (na podstawie przepisów o sygnalizacji), $\boldsymbol{S}$={*S1, S2, S3, …*}

$\boldsymbol{S_e}$ – zbiór wskazań e-tego sygnalizatora.

Znajduje to odzwierciedlenie w funkcjach realizowanych przez układ nastawczo-kontrolny i współpracujące z nim układy. W ramach opracowywania aplikacji systemu ETCS do kodera LEU wgrywane jest oprogramowanie zawierające stałą informacje o tym zbiorze dopuszczalnych wskazań określonego sygnalizatora, z którego obwodem nastawczo-kontrolnym koder LEU ma współpracować. Każdemu wykrytemu przez koder LEU wskazaniu odpowiadającemu określonemu wskazaniu ze zbioru jest przyporządkowany jednoznacznie jeden telegram wysyłany do balisy przełączalnej. Telegramy te są opracowane zgodnie z zasadami przyjętymi dla ETCS i na podstawie cech charakteryzujących warunki realizacji określonego przebiegu lub grupy przebiegów. Wśród nich znajduje się m.in. profil prędkości.

Oprócz tych telegramów istnieją jeszcze telegramy nadawane w przypadkach usterki fragmentu lub całości kodera oraz wykrycia sygnału wskazywanego przez sygnalizator, który nie odpowiada sygnałom ze zbioru sygnałów dopuszczalnych dla danego sygnalizatora. Telegramy te są telegramami nakazującymi zatrzymanie pojazdu.

Należy podkreślić to, że dobór telegramów na podstawie wskazań jest:

- Ustalony na etapie projektowania aplikacji ETCS,
- Wynika z większej liczby zmiennych niż opisane w dokumentach formalnych stopnie prędkości przekazywane przez sygnalizator na kolejnych drogach jazdy.

Opis warstwy powinien obejmować jednoznaczne przyporządkowanie określonego telegramu do określonego sygnału, np.:

- S1 – telegram 1.,
- S2 – telegram 2.,
- S10a – telegram 3.,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- S10a ze wskaźnikiem W24 (wskaźnik oznaczający wyjazd na tor szlaku dwutorowego lub wielotorowego w kierunku przeciwnym do zasadniczego [32]) – telegram 4.

1.7 Inicjatywy ujednolicenia interfejsów urządzeń srk

Opis interfejsów w systemach sterowania ruchem kolejowym stanowi przedmiot wielu prac naukowych i badawczych. Wyróżnić należy inicjatywę EULYNX [39], celem której jest definiowanie i standaryzacja interfejsów dla przyszłych cyfrowych systemów automatyki, transmisji i sterowania ruchem kolejowym przy istotnym ograniczeniu kosztów cyklu życia systemów srk. W inicjatywie tej zdefiniowano dedykowaną metodę normalizacji. Dane wejściowe od uczestników takich jak zarządcy infrastruktury czy producenci urządzeń są gromadzone, ustrukturyzowane i klasyfikowane. Eksperci w odpowiednich dziedzinach przedstawiają architekturę systemu i wymagania systemowe, dzięki czemu cele redukcji kosztów poprzez digitalizację i zgodność z istniejącymi podsystemami „Sterowanie" można osiągnąć poprzez wspólny przegląd i weryfikację. EULYNX zapewnia standardowe interfejsy i zdefiniowane kategorie funkcji mających na celu integrację komponentów plug-and-play w systemie sterowania ruchem [22][45].

EULYNX został rozwinięty z projektu DB Netz AG (niem. Die Deutsche Bahn Netz Aktiengesellshaft) NeuPro (niem. Neue Produktionsverfahren). NeuPro od 2012 r. dąży do stworzenia znormalizowanych specyfikacji interfejsu w krajowych aplikacjach niemieckich. W inicjatywie EULYNX bierze udział wielu producentów systemów srk oraz zarządców infrastruktury z wybranych europejskich krajów.

W ramach projektu NCBiR (Narodowe Centrum Badań i Rozwoju) i PKP PLK S.A. pt.: „Standaryzacja wybranych interfejsów komputerowych urządzeń i systemów sterowania ruchem kolejowym (srk)" realizowanego przez Instytut Kolejnictwa i firmę Rail-Mil Computers Sp. z o.o. sp. k. opracowywane są standardy wybranych interfejsów pomiędzy urządzeniami srk [74].

1.8 Metody badania układów przekaźnikowych

Układy przekaźnikowe wciąż stanowią istotną część sterowników na kolei, a szerzej w przemysłowych systemach sterowania. Opracowano metody weryfikacji takich układów. Najpowszechniej stany układów i przejścia między stanami są wyrażane jako wyrażenia boolowskie, na przykład przy pomocy grafów BDD (ang. binary decision diagram – binarny diagram decyzyjny), a potem weryfikowane przy pomocy weryfikatorów takich jak Spin [25] (ang. Simple Promela Interpreter) czy NuSMV [25]. Logika układów może być też wyrażana

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

jako diagramy stanów UML [24] (ang. Unified Modeling Language – zunifikowany język modelowania). Ta ostatnia praca dotyczy badania tylko sytuacji kolizji pociągów.

W pracy [26] użyto specyfikacji RAISE (ang. Rigorous Approach to Industrial Software Engineering) i weryfikatora SAL (ang. Symbolic Analysis Laboratory), a w pracy [28] autorzy budują bezpośrednio strukturę Kripke i weryfikują własności przy pomocy SONOLAR SMT (ang. Solver for non-linear Arithmetic, SMT - satisfiability modulo theories) w ograniczonej weryfikacji modelowej. W pracy [27] użyto specyfikacji Railway Network DSL (ang. Domain-Specific Language) i ponownie ograniczonej weryfikacji modelowej w RT-Tester (RT – ang. real time). NORMA [3] ma wbudowany edytor graficzny układów, a do weryfikacji używa Timed SMV. W artykule [20] przedstawiono specyfikację w B-Method i weryfikację przy pomocy ProB.

W pracach [44][11] użyto automatów czasowych do modelowania układu, oraz weryfikacji przy pomocy Uppaal [7]. Do weryfikacji zastosowano automaty-obserwatorów, podobnie jak w metodzie przyjętej dalej w rozprawie. Jednak nadal wymagane są formuły temporalne, tyle że zastosowane do obserwatorów a nie do samego układu.

W pracy [37] do analizy pracy obwodów przekaźnikowych zastosowano język opisu sprzętu VHDL (ang. Very High Speed Integrated Circuits Hardware Description Language). Symulacja funkcjonalna z wykorzystaniem symulatora logicznego została przeprowadzona w pakiecie Active_HDL. Modelowano pracę zestyków i czasy ich przełączania.

Stosowane w praktyce projektowej metody badania układów przekaźnikowych bazują na doświadczeniu i wiedzy projektanta. W procesie projektowania i weryfikacji prawidłowości rozwiązań wykorzystuje się zbiór dokumentów zawierających sformułowane zasady i/lub gotowe, sprawdzone rozwiązania. Przykładem zasady jest kontrolowanie pracy przekaźników klasy C (wynikające z zapisu standardu UIC [61] i norm IEC [54][56][3]) poprzez włączenie w inny obwód, umożliwiający tę kontrolę, zestyku rozwiernego kontrolowanego przekaźnika [78]. Przykładem gotowego rozwiązania jest układ kontroli prawidłowości pracy impulsatora [2], wykorzystywanego do generowania na sygnalizatorze przytorowym migającego światła zielonego.

[3] UIC to Międzynarodowy Związek Kolei (fr. Union Internationale des Chemins de fer) – organizacja zrzeszająca przedsiębiorstwa zajmujące się transportem kolejowym i reprezentująca je na arenie międzynarodowej. IEC to Międzynarodowa Komisja Elektrotechniczna (ang. International Electrotechnical Commission) –organizacja globalna opracowująca i publikująca międzynarodowe normy.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Przyjęcie rozwiązania gotowego (albumowego) zazwyczaj powoduje brak głębszego sprawdzania jego poprawności, ponieważ zakłada się, że idea zawarta we wzorcu (albumie) została już kiedyś zweryfikowana i dogłębnie sprawdzona. Jest to podejście stosowane szeroko w praktyce ale ułomne. Najczęściej dostępny jest sam wzorzec, często bez opisu, a jeśli taki opis istnieje, to może nie zawierać sformułowanych warunków, które należy spełnić, aby wzorzec mógł zostać zastosowany (p.. w albumie blokady Eap nie sformułowano warunku konieczności opóźnienia zmiany stanu przekaźników Kc i Kpz kontroli świateł semafora wjazdowego współpracującego z zestawem blokady, w celu prawidłowej pracy przekaźników W i Z kontrolujących wjazd pociągu na stację)[4].

Jednocześnie w trakcie sprawdzania poprawności przyjętego rozwiązania odbiegającego od standardowego, określonego w zatwierdzonym opracowaniu (albumie), wykorzystuje się analizę FMEA (ang. Failure mode and effects analysis – analiza rodzajów i skutków możliwych błędów [60]) [53]. Do jej wykonania niezbędna jest wiedza o działaniu elementów, układów i systemów objętych analizowanym projektem (zespół wykonujący analizę korzysta w tym zakresie z dokumentacji technicznej: DTR (dokumentacja techniczno-ruchowa [71]), opisów, rysunków, specyfikacji) a także przytoczonych wcześniej zasad projektowania (i zawierających je dokumentów). Wykonywana analiza FMEA często polega na ręcznym prześledzeniu działania obwodów przy normalnej pracy i wystąpieniu usterek. Jest ona czasochłonna i nie uniemożliwia pominięcia niektórych sprawdzeń. Przy wykryciu nieprawidłowości działania, w konsekwencji czego wprowadzono zmianę w projekcie, część sprawdzeń należy powtórzyć a do tego celu zidentyfikować wpierw, które wymagają powtórzenia.

[4] Skróty Kc, Kpz wynikają z funkcji realizowanych przez przekaźniki. Nie są związane z typem ani odmianą przekaźnika.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

1.9 Cel i teza rozprawy

Opracowanie specyfikacji interfejsu wymaga zadbania o prawidłowe działanie współpracujących urządzeń i powiązania. Zagadnienie dotyczy wielu warstw. W pracy rozważona zostanie warstwa logiczna, w której sygnały mogą zostać określone jako binarne. Metoda powinna obejmować opracowanie modeli współpracujących urządzeń uwzględniających ich cechy istotne z punktu widzenia rozważanej warstwy i potencjalnych usterek elementów tych urządzeń. Rozważone zostaną powiązania urządzeń przekaźnikowych z komputerowymi, ponieważ eksploatowane są różne rozwiązania urządzeń komputerowych, których współpraca z innymi urządzeniami (wykonanymi w tej samej technologii lub starszej) może być i jest realizowana poprzez interfejsy równoległe z wykorzystaniem przekaźników jako elementów pośredniczących.

Sformułowano cel pracy:

Celem pracy jest przedstawienie metody pozwalającej wykazać poprawność działania i współdziałania przekaźnikowych i komputerowych urządzeń sterowania ruchem kolejowym, lub wykryć błędy w przypadku ich niepoprawnej konstrukcji.

Do osiągnięcia tak sformułowanego celu wymagane jest opracowanie modeli elementów urządzeń srk sprawnych i uszkodzonych, określenie narzędzia do modelowania, określenie zasad opracowywania modelu z elementów i weryfikacji za pomocą narzędzia interesujących cech modelu.

Sformułowano następującą tezę pracy:

Zastosowanie weryfikacji temporalnej do badania komputerowych i przekaźnikowych układów sterowania pozwala wykazać poprawność ich działania i współdziałania, lub zidentyfikować błędy projektowania tych układów i ich powiązań.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

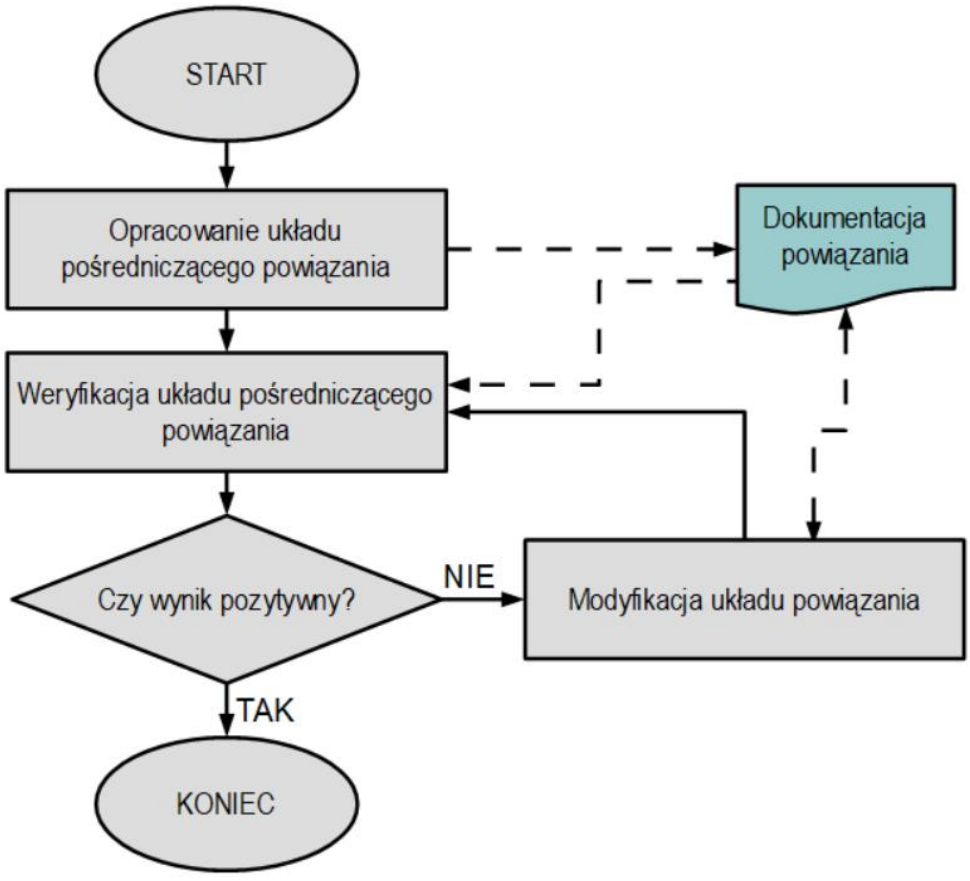
1.10 Zarys proponowanej metody

Powiązane systemy sterowania ruchem kolejowym i układ pośredniczący najczęściej są zrealizowane przy wykorzystaniu cyfrowych wejść i wyjść równoległych, których kombinacja oznacza określony komunikat (polecenie, meldunek) i pośrednio zawierający informację o stanie urządzenia nadającego go. Proponowana metoda zakłada, że dwa urządzenia, które mają zostać powiązane, mają elementy kończące się wyjściami cyfrowymi. Wyjścia kończą się przekaźnikami.

Proponowana metoda może zostać przedstawiona jako sekwencja następujących działań:

1) Opracowanie układu pośredniczącego powiązania.
2) Weryfikacja układu pośredniczącego powiązania.
3) W przypadku niepowodzenia (zakleszczenia lub braku terminacji), na podstawie wyniku symulacji (analizy kontrprzykładu) należy wprowadzić modyfikację w układzie powiązania i wykonać ponownie weryfikację.
4) Zakończona pozytywnie weryfikacja kończy proces.

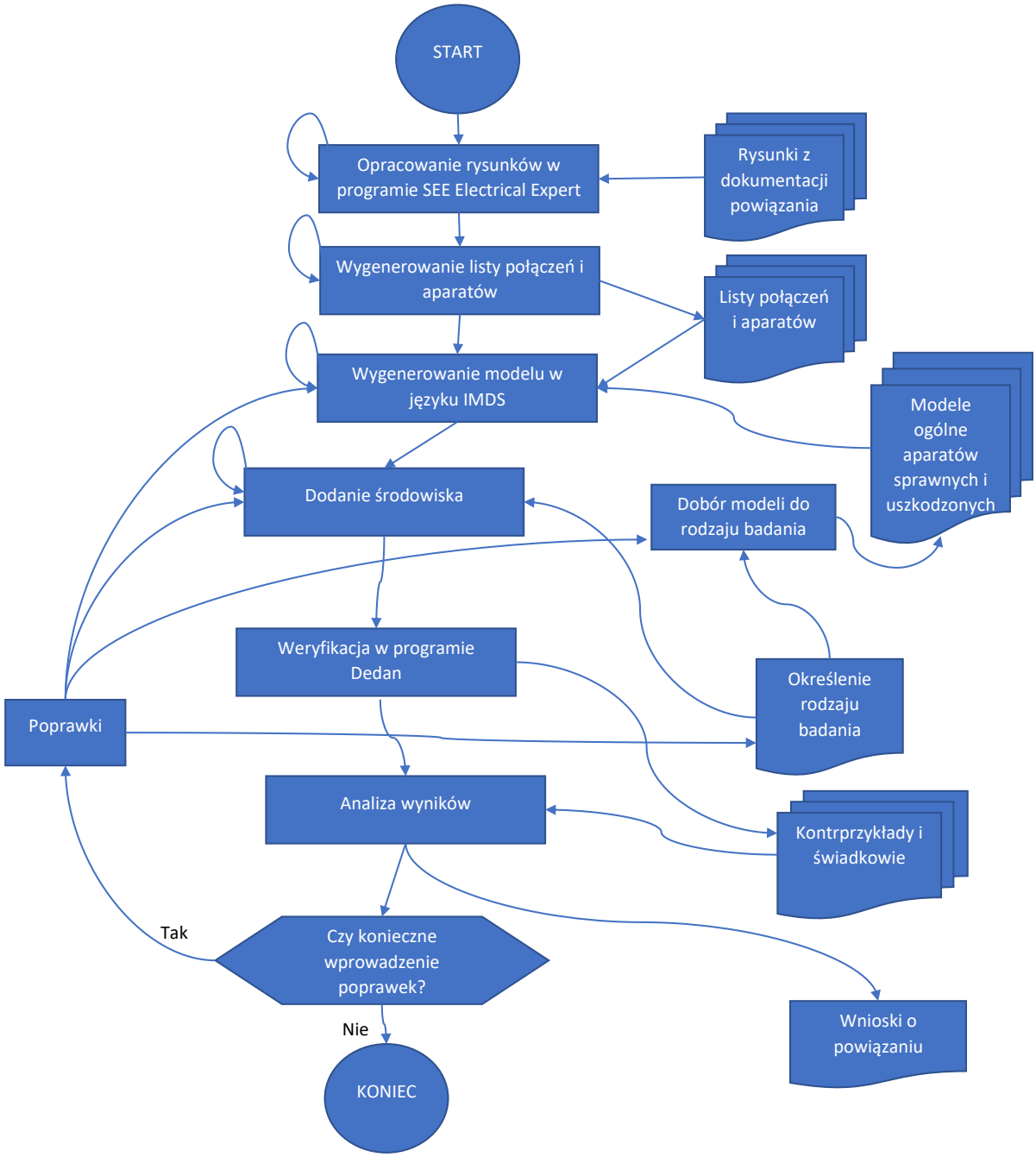
Sekwencję tę przedstawiono graficznie na rysunku 5.



Rysunek 5. Graficzne przedstawienie metody. Źródło: opracowanie własne.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Czynności opisane w punktach 3) i 4) mogą zostać przedstawione dokładniej w postaci grafu przedstawionego na rysunku 6.



Rysunek 6. Uszczegółowiony graf weryfikacji. Źródło: opracowanie własne

Poszczególne czynności i pojęcia zostały szerzej omówione w kolejnych rozdziałach rozprawy: program Dedan, pojęcia świadków i kontrprzykładów w rozdziale 2.4, język IMDS w rozdziale 2.5, określenie rodzaju badania w rozdziałach 2.2, 6.1 i 6.6.3, określenie środowiska w rozdziale 6.6.4, rysunki i zestawienia opracowane w programie SEE w rozdziale 6.6.2, modele aparatów w rozdziale 4.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

2 Weryfikacja formalna systemów

2.1 Wprowadzenie

Weryfikacja formalna to postępowanie mające na celu udowodnienie lub obalenie poprawności zamierzonych sieci logicznych leżących u podstaw systemu w odniesieniu do pewnej formalnej specyfikacji lub własności, przy użyciu formalnych metod matematyki.

Weryfikacja formalna może być pomocna w dowodzeniu poprawności systemów takich jak układy cyfrowe z pamięcią wewnętrzną oraz oprogramowanie wyrażone w postaci kodu źródłowego.

Weryfikacja tych systemów odbywa się poprzez dostarczenie formalnego dowodu na abstrakcyjnym modelu matematycznym systemu, przy czym zgodność między modelem matematycznym a naturą systemu jest skądinąd znana poprzez konstrukcję, chociaż zawsze robi się przy tym pewne uproszczenia. Przykładami obiektów matematycznych często wykorzystywanych do modelowania systemów są: automaty skończone, systemy przejść etykietowanych (LTS), sieci Petriego, systemy dodawania wektorów, automaty czasowe, automaty hybrydowe, algebra procesów, formalna semantyka języków programowania, taka jak semantyka operacyjna, semantyka denotacyjna, semantyka aksjomatyczna i logika Hoare'a.

Metody formalne pozwalają badać podstawową funkcję – realizację przez systemy określonej sekwencji związanej z ich pracą.

Można wyróżnić następujące metody:

- Weryfikacja modelowa (model checking),
- Weryfikacja dedukcyjna (deductive verification).

Weryfikacja modelowa polega na systematycznie wyczerpującej eksploracji modelu matematycznego (jest to możliwe dla modeli skończonych, ale także dla niektórych modeli nieskończonych, gdzie nieskończone zbiory stanów mogą być efektywnie reprezentowane w sposób skończony poprzez zastosowanie abstrakcji lub wykorzystanie symetrii). Zazwyczaj polega to na eksploracji wszystkich stanów i przejść w modelu, poprzez zastosowanie inteligentnych i specyficznych dla danej dziedziny technik abstrakcji, aby rozważyć całe grupy stanów w jednej operacji i zmniejszyć czas obliczeń. Właściwości, które mają być weryfikowane, są często opisywane w logikach temporalnych, takich jak linear temporal logic

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

(LTL), Property Specification Language (PSL), SystemVerilog Assertions (SVA) lub computational tree logic (CTL).

Wielką zaletą weryfikacji modelowej jest to, że często jest ona w pełni automatyczna, po manualnym wyspecyfikowaniu własności i częściowo manualnym zbudowaniu modelu. Jej podstawową wadą jest natomiast to, że generalnie nie skaluje się do dużych systemów – modele symboliczne są zwykle ograniczone do kilkuset stanów, podczas gdy jawne wyliczanie stanów wymaga, aby badana przestrzeń stanów była stosunkowo mała. Wymyślono wiele metod redukcji przestrzeni stanów i weryfikacji symbolicznej.

Innym podejściem jest weryfikacja dedukcyjna. Polega ona na wygenerowaniu z systemu i jego specyfikacji (oraz ewentualnie innych adnotacji) zbioru matematycznych zobowiązań dowodowych, których prawdziwość implikuje zgodność systemu z jego specyfikacją, oraz wywiązaniu się z tych zobowiązań przy pomocy systemów wspomagających dowodzenie twierdzeń (takich jak HOL, ACL2, Isabelle, Coq czy PVS), lub automatycznego dowodzenia twierdzeń, w tym w szczególności solwerów satisfiability modulo theories (SMT). Podejście to ma tę wadę, że może wymagać od użytkownika szczegółowego zrozumienia, dlaczego system działa poprawnie, i przekazania tej informacji systemowi weryfikacji albo w postaci ciągu twierdzeń do udowodnienia, albo w postaci specyfikacji (niezmienników, warunków wstępnych, warunków końcowych) składników systemu (np. funkcji lub procedur) i być może podskładników (takich jak pętle lub struktury danych).

Weryfikacja modelowa pozwala na dowody poprawności zachowania systemów w czasie. Bardzo ważne jest rozróżnienie weryfikacji od testowania. Weryfikacja modelowa nie sprawdza konkretnych przebiegów systemu, lecz analizuje je „wszystkie razem", dając formalny dowód spełnienia lub niespełnienia określonych własności.

Ważne jest, aby dobrać taką metodologię weryfikacji, żeby z jednej strony jak najlepiej odzwierciedlić cechy weryfikowanego systemu, a z drugiej ułatwić wielokrotną weryfikację wielu wersji przygotowywanego projektu.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

2.2 Weryfikacja temporalna

Weryfikacja temporalna (modelowa) polega na sprawdzeniu występowania lub niewystępowania określonych cech w sekwencji zachowania się badanego układu. Weryfikacja taka ma sens w przypadku występowania w układzie elementów pracujących współbieżnie, ponieważ sekwencyjna praca elementów układu (nawet w przypadku występowania w niej niedeterminizmu) może być łatwo prześledzona manualnie. Złożenie równoległych aktywności wielu elementów daje tak dużą liczbę możliwych zachowań, że nie sposób ich prześledzić manualnie. Weryfikacja temporalna sprawdza wszystkie możliwe zachowania układu, dając dowód określonego działania. Weryfikacja temporalna jest uznanym sposobem weryfikacji systemów stanowych, które działają w środowisku sekwencji sygnałów z otoczenia i same generują takie sekwencje.

Paralelizm zadań w systemie współbieżnym stwarza trudności spowodowane możliwie nieskończonym zbiorem ich wspólnych wykonań. Współczesne systemy dodają do równoległości nowy wymiar trudności: niezależne węzły współpracujące w środowisku rozproszonym. Poziom równoległości jest znacznie wyższy niż w systemach scentralizowanych, a w takim systemie nie istnieje prawdziwy globalny stan obliczeń.

W systemie współbieżnym nie istnieją żadne wspólne zmienne, a jedynym sposobem komunikacji między zadaniami jest przekazywanie wiadomości przez sieć. Serwer nie może nawet oczekiwać, że odpowiedź na wysłany komunikat kiedykolwiek nadejdzie. Z wyjątkiem ściśle sprzężonych systemów ze wspólną magistralą, komunikaty są przekazywane między serwerami w sposób asynchroniczny. W systemie rozproszonym nie ma ani stanu globalnego, ani nielokalnego.

Powiązanie dwóch lub większej liczby urządzeń srk jest właśnie przypadkiem, w którym każde z wiązanych urządzeń pracuje współbieżnie względem pozostałych. Ich praca nie jest co do zasady synchronizowana. Również w mniejszej skali, przekaźnika, zestyki określonego rodzaju (np. zwierne), pomimo wymaganej regulacji [51], przełączają się nie jednakowo dla poszczególnych egzemplarzy danego typu (to znaczy np. w jednym egzemplarzu przy wzbudzaniu najpierw zestyk numer 1 a potem 2, a w drugim najpierw 2 a potem 1).

Do automatyzacji procesu weryfikacji może posłużyć formalizm Integrated Model of Distributed Systems [17] (IMDS). Jego unikatowe cechy wspomagające automatyczną weryfikację to:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- ograniczony zestaw badanych cech, ale weryfikowanych automatycznie, bez specyfikowania formuł temporalnych; są to zakleszczenia i terminacja;
- znajdowanie częściowych zakleszczeń i terminacji, dotyczących fragmentów układu lub nawet jego pojedynczych elementów; podczas gdy pozostała część układu może nie doświadczać skutków częściowego zakleszczenia/terminacji; większość weryfikatorów znajduje tylko zakleszczenia/terminację totalne, a o częściowe użytkownik musi spytać formułując odpowiednie zdania temporalne;
- formalizm graficzny do modelowania ma identyczną semantykę jak model algebraiczny do weryfikacji, nie ma więc potrzeby translacji z opisu układu do języka weryfikacji [16];
- sprawiedliwość weryfikacji[5] (większość weryfikatorów jest niesprawiedliwych lub słabo sprawiedliwych, co prowadzić może do wykrycia nieistniejących zakleszczeń [15]).

Weryfikacja modelowa w formalizmie IMDS odpowiada na pytania o:

- zakleszczenie,
- nieuchronną terminację procesu,
- możliwą terminację procesu.

Cechy te mogą dotyczyć określonego procesu lub grupy procesów.

Metoda z wykorzystaniem IMDS pozwala wykazać prawidłowość działania układu, oraz zachowanie bezpieczeństwa przez układ zarówno przy niepoprawnej sekwencji sygnałów na wejściu jak i przy uszkodzeniu pojedynczych elementów.

Prawidłowość działania układu musi zostać każdorazowo określona dla badanego układu. W szczególności prawidłowe działanie oznaczać może wzbudzenie, a następnie odwzbudzenie pewnych przekaźników w układzie w wyniku otrzymania sekwencji sygnałów świadczących o ruchu pociągu, wydania poleceń od personelu sterowania ruchem bądź zmiany stanu kontrolowanych elementów lub realizacji wewnętrznej funkcji układu (np. odmierzenia upływu czasu).

[5] Pojęcie sprawiedliwości wyjaśniono w rozdziale 2.9.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Podstawowe badanie w metodzie dotyczy nieuchronności poprawnego zadziałania przy poprawnej sekwencji sygnałów wejściowych. Inne badanie weryfikuje, czy układ zachowuje się bezpiecznie, gdy nie otrzymuje sygnałów wejściowych we właściwej kolejności.

Kolejnym badaniem jest sprawdzenie, czy w przypadku prawdopodobnego uszkodzenia układ również zachowuje się bezpiecznie. Oczywiście układ może zostać uszkodzony na wiele sposobów i w przypadku weryfikacji wykonywanej szeroko, np. na potrzeby certyfikacji nowego typu urządzenia [70][64], należałoby zbadać wszystkie możliwe uszkodzenia, a tam, gdzie to konieczne [53][78], również jednoczesne wystąpienia wielu uszkodzeń.

2.3 Zintegrowany model systemów rozproszonych

Formalizm IMDS [17] opiera się na zdefiniowaniu konfiguracji badanego systemu. Konfiguracja to zbiór stanów powiązanych urządzeń w określonej chwili czasu. Nie używa się słowa „stan", ponieważ jest ono zarezerwowane dla pewnego elementu składowego konfiguracji.

System składa się z serwerów (rozproszonych, niezależnych węzłów), oraz agentów reprezentujących rozproszone obliczenia (operacje realizowane na serwerach i pomiędzy nimi). Agenci przemieszczają się pomiędzy serwerami przy pomocy komunikatów. Konfiguracja jest zbiorem stanów wszystkich serwerów i komunikatów wszystkich agentów, z wyjątkiem agentów, którzy skończyli swoją pracę. Można zastosować oznaczenia:

$M = \{m_1, m_2, \dots\}$ – skończony zbiór komunikatów (agentów),

$P = \{p_1, p_2, \dots\}$ – skończony zbiór stanów (serwerów).

Działanie systemu jest opisywane przez przejścia od konfiguracji do konfiguracji, które są nazywane akcjami. Akcja „zabiera" z konfiguracji parę (*stan serwera*, *komunikat agenta*) tworzy nową konfigurację, w której stan jest zastępowany przez nowy stan tego samego serwera, a komunikat jest zastępowany przez nowy komunikat tego samego agenta, adresowany do tego samego lub innego serwera. Wyjątkiem jest akcja terminująca agenta, która produkuje nowy stan, ale nie produkuje nowego komunikatu.

Relacja akcji Λ jest binarną relacją na zbiorach M i P:

$$\Lambda \subset (M \times P) \times (M \times P) \tag{9}$$

Dla elementów Λ używana jest notacja:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

$$\lambda = \big((m, p), (m', p')\big) \in \Lambda \tag{10}$$

Akcja może zostać zapisana jako:

$$(m, p)\Lambda(m', p') \tag{11}$$

Wykonanie akcji (żądanej w komunikacie) powoduje zmianę stanu serwera i wydanie przez agenta kolejnego komunikatu. Wykonanie akcji przekształca zatem aktualny stan i oczekujący komunikat w następny stan i następny komunikat.

Dla akcji $(m, p)\Lambda(m', p')$ można powiedzieć, że:

- Para (m, p) pasuje, tj. wiadomość m może wyzwolić akcję wspólnie z p,
- (m, p) jest parą wejściową,
- (m', p') jest parą wyjściową,
- Stan p jest bieżący,
- Wiadomość m jest oczekująca, jest to bieżąca wiadomość danego agenta,
- p' jest następnym stanem,
- m' jest następną wiadomością.

Konfiguracja początkowa systemu T_{ini} składa się z początkowych stanów wszystkich serwerów i startowych komunikatów wszystkich agentów.

$T \subset P \cup M$ – konfiguracja

$T_{ini} = P_{ini} \cup M_{ini}$ – konfiguracja początkowa

P_{ini} – zbiór stanów początkowych; $P_{ini} \subset P$

M_{ini} – zbiór wiadomości początkowych; $M_{ini} \subset M$

System rozpoczyna pracę z początkowych stanów wszystkich serwerów (jeden stan dla każdego serwera) i początkowych wiadomości wszystkich agentów (jedna wiadomość dla każdego agenta).

Zachowanie systemu opisanego w IMDS jest opisane przez Labeled Transition Systems (LTS [63]), reprezentowane przez etykietowane grafy skierowane.

Dla LTS reprezentującego zachowanie systemu z wykorzystaniem IMDS, węzłami są konfiguracje systemu, czyli zbiory stanów bieżących elementów zbioru P i komunikatów oczekujących. Termin "stan" jest zarezerwowany dla elementów zbioru P. Korzeniem jest konfiguracja początkowa T_{ini}. Zbiór etykiet to zbiór akcji Λ systemu. Serwery zaczynają od

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

P_{ini}, a agenci od M_{ini}; dlatego konfiguracja początkowa jest sumą ich zbiorów. Zbiór łuków skierowanych jest określony przez akcje, które przekształcają swoje konfiguracje wejściowe na swoje konfiguracje wyjściowe w następujący sposób:

$$\forall_{\lambda\in\Lambda}\lambda = \big((m,p),(m',p')\big); T_{inp}(\lambda) \supset \{m,p\} \Rightarrow T_{out}(\lambda) = \big(T_{inp}(\lambda) \setminus \{m,p\}\big)\cup\{m',p'\} \tag{12}$$

gdzie:

$T_{inp}(\lambda)$ – konfiguracja wejściowa akcji λ,

$T_{out}(\lambda)$ – konfiguracja wyjściowa akcji λ.

$LTS = \langle N, n_0, W\rangle$

Gdzie:

N - zbiór węzłów, $N = \{n_0, n_1, \dots\}$

n_0 – korzeń, $n_0 \in N$, $n_0 = T_{ini}$

W – zbiór skierowanych, etykietowanych przejść, $W \subset N \times \Lambda \times N$, $W = \left\{\left(T_{imp}(\lambda), \lambda, T_{out}(\lambda)\right) | \lambda \in \Lambda\right\}$

Semantyka systemu jest z przeplotem, czyli, że jedna akcja jest wykonywana w określonym momencie. Z tego powodu pojedyncze akcje są etykietami w LTS. Wykonanie akcji przekształca konfigurację wejściową w wyjściową w taki sposób, że para wejściowa (*wiadomość*, *stan*) jest usuwana i zastępowana parą wyjściową (*następna wiadomość, następny stan*). Wiadomości i stany nie uwzględnione w parze wejściowej pozostają niezmienione. LTS zawiera wszystkie możliwe przebiegi systemu jako ścieżki w grafie.

Ponieważ akcja jest wykonywana na serwerze, w każdej akcji komunikat wejściowy musi być skierowany do serwera wyznaczonego przez stan wejściowy.

Następny stan generowany w akcji wyznacza ten sam serwer, co stan wejściowy. Zapewnia to ciągłość działania serwera. Podobnie, następny komunikat wygenerowany w akcji wyznacza tego samego agenta, co komunikat wejściowy. Zapewnia to ciągłość zachowania agenta.

Formalna definicja IMDS na potrzeby ekstrakcji procesów przedstawia się następująco:

$S = \{s_1, s_2, \dots\}$ – skończony zbiór serwerów,
$A = \{a_1, a_2, \dots\}$ – skończony zbiór agentów,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

$P(s) = \{p_{1s}, p_{2s}, \dots\}, s \in S$ – skończony zbiór stanów serwera s,
$M(a) = \{m_{1a}, m_{2a}, \dots\}, a \in A$ – skończony zbiór wiadomości agenta a,
$M(s) = \{m_{1s}, m_{2s}, \dots\}, s \in S$ – skończony zbiór wiadomości skierowanych do serwera s.

Dla elementów zbiorów $P(s)$, $M(a)$ i $M(s)$, można powiedzieć, że element wyznacza odpowiednio odpowiadający mu serwer, agent i serwer.

Wyodrębnienie procesów wymaga pewnych ograniczeń na zbiorach P, M i relacji Λ:

- Każdy stan musi wyznaczyć jakiś serwer:

$$P = \bigcup_{s \in S} P(s), \forall s_1, s_2 \; s_1 \neq s_2 \Rightarrow P(s_1) \cap P(s_2) = \emptyset \tag{13}$$

- Każda wiadomość musi wyznaczyć jakiegoś agenta i musi być skierowana do jakiegoś serwera:

$$M = \bigcup_{a \in A} M(a) = \bigcup_{s \in S} M(s), \forall a_1, a_2 \; a_1 \neq a_2 \Rightarrow M(a_1) \cap M(a_2) = \emptyset, \forall s_1, s_2 \; s_1 \neq s_2 \Rightarrow M(s_1) \cap M(s_2) = \emptyset \tag{14}$$

- Stan wejściowy akcji i komunikat wejściowy muszą wyznaczać ten sam serwer; stany wejściowy i wyjściowy muszą wyznaczać ten sam serwer; komunikaty wejściowy i wyjściowy muszą wyznaczać tego samego agenta:

$$\forall \lambda \in \Lambda \; \lambda = \big((m, p), (m', p')\big), m \in M(a) \Rightarrow m' \in M(a), p \in P(s) \Rightarrow \big(p' \in P(s) \wedge m \in M(s)\big) \tag{15}$$

- początkowy zbiór stanów zawiera dokładnie jeden stan dla każdego serwera:

$$\forall s \in S \; card(P(s) \cap P_{ini}) = 1 \tag{16}$$

- początkowy zbiór komunikatów zawiera dokładnie jeden komunikat dla każdego agenta:

$$\forall a \in A \; card(M(a) \cap M_{ini}) = 1 \tag{17}$$

Dodano nowy typ akcji terminującej agenta, w której nie ma nowego komunikatu:

$$\Lambda \subset (M \times P) \times (M \times P) \cup (M \times P) \times (P) \tag{18}$$

Stąd akcja Λ nie jest ściśle relacją, ponieważ zawiera zarówno czwórki, jak i trójki. Bardziej formalnie, do M można dodać wiadomość "agent termination", która jest zabroniona na wejściu każdej akcji.

Takie określenie akcji Λ wymaga rozbudowy ograniczeń w następujący sposób:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

$$\forall \lambda \in \Lambda\ \lambda = \big((m,p),(p')\big), p \in P(s) \Rightarrow \big(p' \in P(s) \wedge m \in M(s)\big) \tag{19}$$

$$\forall_{\lambda \in \Lambda} \lambda = \big((m,p),(p')\big); T_{inp}(\lambda) \supset \{m,p\} \Rightarrow T_{out}(\lambda) = \big(T_{inp}(\lambda) \setminus \{m,p\}\big) \cup \{p'\} \tag{20}$$

Asynchroniczne procesy sekwencyjne w systemie są zdefiniowane jako sekwencje działań.

Dwoma elementami wyjściowymi dowolnej akcji (lub jednym elementem wyjściowym dla akcji kończącej działanie agenta) mogą być elementy wejściowe innych akcji. Jeśli wybierzemy stany jako nośnik procesu, to otrzymamy *proces serwera* przebiegający wraz ze zmieniającymi się stanami serwera. Stany pomiędzy akcjami definiują *relację następstwa* pomiędzy akcjami. Z tego punktu widzenia komunikaty pośredniczące pomiędzy akcjami są przekazywane pomiędzy akcjami różnych procesów serwerów, czyli służą jako środek komunikacji. Procesy serwerów nie kończą się (w sensie akcji kończącej); stan wyjściowy jest obecny w każdej akcji.

Z drugiej strony można wybrać komunikaty do akcji wykonywanych w ramach określonego procesu. Z tej perspektywy komunikaty agenta tworzą *proces agenta*. Komunikaty pomiędzy akcjami definiują *relację następstwa* pomiędzy akcjami. Proces przebiega jako sekwencja komunikatów, a odwiedzane stany serwerów są środkiem komunikacji agentów. Ponieważ mogą istnieć akcje bez komunikatu wyjściowego, procesy agentów mogą zostać przerwane.

Dla procesu serwera s: pierwsza akcja procesu ma na wejściu początkowy stan serwera, wszystkie stany, przez które może przejść proces wyznaczają serwer s:

$$B(s) = \langle \lambda_0, \lambda_1, \dots \rangle | \forall_{i=0,1,\dots} \lambda_i \in \Lambda \tag{21}$$

$B(s)$ – proces serwera

$$\lambda_0 = \big((m,p),(m_0,p_0)\big) \vee \lambda_0 = \big((m,p),(p_0)\big), p \in P_{ini} \cap P(s) \tag{22}$$

$$\begin{aligned} \forall_{i>0} \lambda_{i-1} = \big((m,p),(m_1,p_1)\big) \vee \lambda_{i-1} = \big((m,p),(p_1)\big), \lambda_i \\ = \big((m_2,p_2),(m_3,p_3)\big) \vee \lambda_i = \big(m_2,p_2,(p_3)\big), p_1,p_2 \in P(s), p_1 = p_2 \end{aligned} \tag{23}$$

Dla procesu agenta a: pierwsza akcja procesu ma na wejściu początkowy komunikat agenta, wszystkie komunikaty, przez które może przejść proces, wyznaczają agenta a:

$$C(a) = \langle \lambda_0, \lambda_1, \dots \rangle | \forall_{i=0,1,\dots} \lambda_i \in \Lambda \tag{24}$$

$C(a)$ – proces agenta

$$\lambda_0 = \big((m,p),(m_0,p_0)\big) \vee \lambda_0 = \big((m,p),(p_0)\big), m \in M_{ini} \cap M(a) \tag{25}$$

$$\begin{aligned} \forall_{i>0} \lambda_{i-1} = \big((m,p),(m_1,p_1)\big), \lambda_i = \big((m_2,p_2),(m_3,p_3)\big) \vee \lambda_i \\ = \big(m_2,p_2,(p_3)\big), m_1,m_2 \in M(a), m_1 = m_2 \end{aligned} \tag{26}$$

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

System zdekomponowany na procesy serwerów jest nazywany widokiem serwerowym, a zdekomponowany na procesy agentów – widokiem agentowym.

Wiele właściwości systemu może być weryfikowanych za pomocą techniki weryfikacji modelowej. W różnych widokach można zaobserwować różne cechy: zakleszczenia komunikacyjne w widoku serwera oraz zakleszczenia dotyczące zasobów i rozproszonej terminacji w widoku agenta.

2.4 Środowisko Dedan

Metody formalne znacznie pomagają, jeśli mogą być stosowane w sposób automatyczny, nie wymagający specjalistycznej wiedzy od użytkowników. W Instytucie Informatyki Politechniki Warszawskiej system Dedan (**De**a**d**lock **an**alyzer) jest wykorzystywany w laboratorium studenckim do modelowania i automatycznej weryfikacji systemów rozproszonych [12]. Studenci modelują swoje rozwiązania w formalizmie IMDS (Integrated Model of Distributed Systems) i sprawdzają je pod kątem zakleszczeń.

Dedan bazuje na logice temporalnej CTL i weryfikacji modelowej [12]. Do wyszukiwania zakleszczeń wykorzystywane są uniwersalne formuły, a proces weryfikacji temporalnej jest ukryty wewnątrz narzędzia. Uniwersalność formuł polega na tym, że nie zależą one od struktury modelu. Program umożliwia identyfikację zakleszczeń całkowitych i częściowych. W przypadku zakleszczenia lub nieskutecznej terminacji przykładowa sekwencja prowadząca do takiej sytuacji jest nazywana **kontrprzykładem** i sporządzana w czytelnej formie grafu przypominającego diagram sekwencji. Można też zbadać nieuchronną lub możliwą terminację agentów, co potwierdza osiągnięcie pożądanej konfiguracji w postaci tzw. **świadka**, który ma postać zbliżoną do kontrprzykładu. Kontrprzykład i świadek są każdy ciągiem stanów serwerów i komunikatów przesyłanych między nimi. Analiza kontrprzykładu w takiej postaci pomaga więc w zlokalizowaniu źródła błędu.

Narzędzie Dedan oferuje dodatkowe funkcje analizy, takie jak obserwacja całkowitego grafu weryfikowanego systemu i symulacja systemu na tym grafie, na kontrprzykładach lub na poszczególnych składowych systemu, gdzie składowe są konwertowane na grafy automatycznie.

Wadą środowiska weryfikacji Dedan jest konieczność specyfikacji modelu w języku wejściowym Dedan. Język ten wywodzi się z formalnej definicji IMDS. Ten styl specyfikacji jest daleki od typowego stylu programowania.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Michalski i Bielecki, autorzy pracy [12], opracowali preprocesor o nazwie Rybu, który przezwycięża tę wadę i upraszcza specyfikację używając języka, o składni zbliżonej do podzbioru języka C. Język Rybu ma znacznie wyższy poziom abstrakcji i jest w stanie generować duże modele z prostych specyfikacji dzięki iloczynowi kartezjańskiemu wartości używanych w kodzie źródłowym serwerów. Ręczne zbudowanie tak dużych modeli jest niemożliwe.

2.5 Implementacja IMDS w języku programowania

W podstawowej definicji IMDS [17] stany i komunikaty zostały zdefiniowane jako proste, niepodzielne byty. W niektórych sytuacjach używa się atrybutów stanów i komunikatów, tworzonych przez wyznaczone serwery i/lub agentów (zbiory $P(s)$, $M(a)$, $M(s)$). W programie Dedan przeznaczonym do weryfikacji systemów rozproszonych, IMDS jest używany jako język specyfikacji. Aby móc definiować typy serwerów i typy agentów oraz deklarować zmienne tych typów, lepiej jest określić stan serwera jako parę (*serwer*, *wartość*). Komunikat może być zdefiniowany jako para (*agent*, *serwer docelowy*).

Agent może jednak wywoływać różne operacje na serwerze. Operacje te będą nazywane "usługami". Komunikaty są więc zdefiniowane jako trójki (*agent*, *serwer*, *usługa*). W takim sformułowaniu komunikat jest wywołaniem usługi serwera przez agenta. Poniżej znajduje się formalna definicja działań w kategoriach stanów jako par i komunikatów jako trójek.
Podstawowe zestawy to serwery, agenci, wartości i usługi:

$S = \{s_1, s_2, \dots\}$ – skończony zbiór serwerów,
$A = \{a_1, a_2, \dots\}$– skończony zbiór agentów,
$V = \{v_1, v_2, \dots\}$– skończony zbiór wartości,
$R = \{r_1, r_2, \dots\}$ – skończony zbiór usług.

Akcja to wykonanie usługi na serwerze w kontekście agenta.

$P \subset S \times V$ – zbiór stanów serwerów,
$M \subset A \times S \times R$ – zbiór wiadomości (kierowanych przez agentów do serwerów w celu wywołania określonej usługi),

W programie Dedan używana jest notacja $s.v$ i $a.s.r$ dla par (s, v) reprezentujących stany serwerów i trójek (a, s, r) reprezentujących wiadomości oraz notacja $\{a.s.r, s.v\} \rightarrow \{a.s'.r', s.v'\}$ dla akcji $\big((a, s, r), (s, v)\big)\Lambda\big((a, s', r'), (s, v')\big)$, przy czym:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

$s, s' \in S, a \in A, r, r' \in R, v, v' \in V, (s, v), (s, v') \in P, (a, s, r)(a, s', r') \in M,$

Notacja w języku programowania jest intuicyjna. Notacja w widoku serwera zawiera następujące elementy:

- Nagłówek (opcjonalny) zawiera nazwę systemu.
- Jako pierwsze definiowane są typy serwerów.
- Następnie definiowane są:
    - serwery,
    - agenci,
    - część inicjalizacyjna.
- Typ serwera składa się z parametrów formalnych określających:
    - używane serwery i agentów,
    - usługi serwera,
    - stany,
    - akcje serwera.
- Stany i usługi mogą być są definiowane w postaci wektorów.
- W części inicjalizacyjnej do każdego serwera przekazywane są parametry aktualne serwerów (będące agentami i serwerami), a po kropce określany jest stan początkowy każdego serwera.
- Dla każdego agenta zdefiniowana jest wiadomość początkowa.

Przedstawione dane wejściowe programu Dedan są zgodne z formalizmem IMDS. Notacja została przedstawiona szerzej w artykule [17].

2.6 Zakleszczenie i terminacja

Metodyka weryfikacji, składająca się ze specyfikacji IMDS [17] i weryfikacji temporalnej [13], pozwala projektantowi na specyfikację systemu rozproszonego w celu łatwego znalezienia różnego rodzaju zakleszczeń (zakleszczenie całkowite/częściowe, zakleszczenie komunikacyjne/zasobowe) oraz rozproszonych terminacji procesów (również całkowitych/częściowych).

Przykładem zakleszczenia serwerów IMDS jest sytuacja, gdy dwa serwery czekają na wiadomości od siebie. Jest to zakleszczenie komunikacyjne, ponieważ sposobem współpracy między procesami serwerów jest przekazywanie komunikatów. Definicja zakleszczenia komunikacyjnego w IMDS to sytuacja, w której na serwerze (przynajmniej jednym) znajdują

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

się oczekujące komunikaty, ale nie mogą być one nigdy obsłużone. Zakleszczenie może dotyczyć wszystkich serwerów (zakleszczenie całkowite) lub podzbioru serwerów (zakleszczenie częściowe); nawet pojedynczy serwer może być zakleszczony.

W ujęciu agentowym zakleszczenie to sytuacja, w której agent ma wiadomość oczekującą na jakimś serwerze, ale nie może zrobić żadnego postępu. Wiadomość nie może zostać obsłużona, ale wiadomości innych agentów być może tak. Ponownie, zakleszczenie może dotyczyć wszystkich agentów (zakleszczenie całkowite) lub podzbioru agentów systemu (zakleszczenie częściowe). Zgodnie z tą definicją, pojedynczy proces może być zakleszczony. Sytuacja taka nazywana jest czasem *zagłodzeniem* (ang. starvation) [67] lub *przeciągnięciem* (ang. stall) [47].

(Częściowa) terminacja w IMDS jest po prostu zniknięciem agenta. Akcja terminująca agenta wiąże parę wejściową (*wiadomość, stan*) z wyjściowym singlem (*następny stan*). Akcja generuje nowy stan serwera, ale nie generuje następnej wiadomości.

Weryfikacja modelowa jest metodą często stosowaną w statycznym wykrywaniu zakleszczeń lub weryfikacji terminacji w systemach rozproszonych [10][5]. Weryfikator modelowy automatycznie wykrywa sytuacje takie jak zakleszczenie lub brak terminacji, używając specyfikacji systemu i formuł temporalnych. Formuły temporalne do weryfikacji zakleszczenia i terminacji są zdefiniowane w kategoriach elementów IMDS (konfiguracje i akcje) w sposób niezależny od kształtu przestrzeni osiągalności danego systemu [13]. Ogólność formuł oznacza, że nie zależą one od specyficznych cech procesów w danym systemie. Jako argumenty wykorzystywane są jedynie cechy procesów serwerów i procesów agentów określone w IMDS. Dwa widoki specyfikacji pozwalają na znalezienie zakleszczeń komunikacyjnych i zakleszczeń na zasobach. W innych formalizmach istnieją niezależne od weryfikowanego systemu formuły pozwalające na znalezienie całkowitego zakleszczenia, ale nie odróżniają one zakleszczenia od terminacji i zwykle dotyczą tylko całkowitego zakleszczenia. Niektóre weryfikatory wykonują automatyczne wykrywanie częściowego zakleszczenia, ale tylko w przypadkach, gdy schemat weryfikowanego systemu jest ściśle ograniczony (na przykład przestrzeń osiągalności czysto cykliczna albo czysto niecykliczna). Podobne ograniczenie dotyczy sprawdzania częściowej terminacji. Jeżeli struktura systemu nie jest ograniczona, to projektant musi posiadać pewną wiedzę z zakresu logiki temporalnej lub równoważnych formalizmów, aby przeprowadzić weryfikację. Szczegółowa analiza technik wykrywania zakleszczeń i sprawdzania zakończenia podana jest w [13].

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Jeśli proces zakleszczy się lub jeśli nie zakończy się, weryfikator modelowy (ang. model checker) tworzy kontrprzykład dla błędu w systemie. Kontrprzykład wspiera projektanta w znalezieniu i poprawieniu błędów w specyfikacji. Kontrprzykład w IMDS jest sekwencją konfiguracji i akcji prowadzących od startu systemu do błędnej konfiguracji.

Formuły niezależne od systemu - używane do wykrywania zakleszczeń i sprawdzania terminacji - zapewniają automatyczną weryfikację. Dlatego formułowanie i weryfikacja właściwości są wykonywane automatycznie wewnątrz programu modelującego Dedan [21], a użytkownik widzi tylko końcowe wyniki. Projektant musi przyswoić sobie formalizm IMDS i język specyfikacji ale nie musi znać logiki temporalnej.

W IMDS zakleszczenie jest definiowane jako sytuacja, w której proces, który nie został zakończony i nie może być kontynuowany (w chwili bieżącej i w przyszłości). Formalne definicje procesów serwerów i procesów agentów różnią się, ponieważ mają one różne cechy.

W procesie serwera mogą wystąpić następujące sytuacje:

- na serwerze znajduje się pasująca para (*komunikat, stan*) – akcja jest przygotowana; proces *biegnie*;
- są oczekujące wiadomości lub nie, ale nie ma pasującej pary na serwerze; pasująca para może się zdarzyć w przyszłości, więc proces serwera *czeka*;
- nie ma żadnych oczekujących wiadomości, a w przyszłości do serwera nie dotrze żadna wiadomość; proces jest *bezczynny*;
- w serwerze są oczekujące wiadomości, ale nie ma pasującej pary, i taka para nie może wystąpić na serwerze w przyszłości; proces jest *zakleszczony*.

Pierwsze trzy przypadki nie są niebezpieczne, ponieważ są typowe dla przebiegu procesu. Na przykład serwery czekają na wywołanie usługi, które nigdy nie nastąpi. Proces serwera nigdy się nie kończy, ale może pozostać bezczynny. Ten ostatni przypadek oznacza zakleszczenie, jeżeli pewne komunikaty oczekują na akceptację przez wykonanie określonych akcji, ale aktualny stan serwera nie odpowiada żadnemu z komunikatów, ani nie będzie odpowiadał w przyszłości. Ponieważ metodą współpracy procesów serwera jest przekazywanie komunikatów, jest to zakleszczenie komunikacyjne.

Sytuacje dotyczące procesu agenta to:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- wiadomość agenta oczekuje na serwerze i pasuje do aktualnego stanu tego serwera - akcja jest przygotowana; agent *działa*;
- wiadomość agenta oczekuje na serwerze, ale nie pasuje do aktualnego stanu tego serwera; pasująca para zawierająca tę wiadomość może pojawić się w przyszłości; agent *czeka*;
- wiadomość agenta oczekuje na serwerze i nigdy nie będzie pasować do żadnego stanu tego serwera - ani teraz, ani w przyszłości; proces jest *zakleszczony;*
- wykonano akcję kończącą agenta, a komunikat agenta nie jest obecny (i nie pojawi się w przyszłości) - proces jest *zakończony.*

Dwa pierwsze przypadki to normalne działanie procesu agenta. Trzeci przypadek oznacza zakleszczenie - komunikat agenta oczekuje na serwerze, ale nigdy nie będzie pasował do żadnego stanu serwera. Jest to zakleszczenie na zasobach, ponieważ procesy agentów komunikują się poprzez zmianę stanów serwerów. Ostatni przypadek oznacza zakończenie pracy agenta, co wynika z definicji akcji kończącej pracę agenta.

2.7 Sprawdzanie modelu w środowisku Dedan

Środowisko Dedan zostało opracowane do praktycznego zastosowania proponowanej metodyki weryfikacji [21]. Program Dedan zawiera wewnętrzny model sprawdzający TempoRG wykorzystujący algorytm CBS (ang. Checking By Spheres) [15][14]. Algorytm weryfikacji został dostosowany do wykorzystania w programie Dedan: został uproszczony do ograniczonego zbioru formuł (dla sprawdzania zakleszczeń i terminacji). Ten weryfikator używa jawnej przestrzeni osiągalności, co pozwala na radzenie sobie z dość małymi specyfikacjami. W przypadku większych specyfikacji wykorzystywane są zewnętrzne weryfikatory modelu: Spin [29], NuSMV [9] oraz Uppaal [7].

LTS układu jako graf jest naturalnie interpretowany jako struktura Kripkego [15]: skończony zbiór *węzłów* - konfiguracje, *węzeł początkowy* - konfiguracja początkowa, całkowita *relacja przejścia* - akcje.

Etykietowanie węzłów polega na przypisaniu zbioru atomowych formuł Boolowskich, które są prawdziwe w danym węźle. Takie etykietowanie jest oczywiste dla konfiguracji IMDS (patrz tabela 3). Dlatego technika weryfikacji modelowej może być sprowadzona do analizy LTS, w szczególności do znajdowania zakleszczeń i sprawdzania terminacji procesów.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Zakleszczenie komunikacji z serwerem definiuje się jako sytuację, w której aktualny stan serwera nie odpowiada żadnemu komunikatowi oczekującemu na serwerze, a serwer nie otrzyma w przyszłości żadnego komunikatu odpowiadającego temu stanowi serwera. W przypadkach wykrywania zakleszczenia komunikacyjnego etykietuje się węzły LTS za pomocą formuł zmiennych logicznych (Boolowskich):

- D_s = prawda, we wszystkich konfiguracjach, w których co najmniej jedna wiadomość jest oczekująca na serwerze s,
- E_s = prawda, we wszystkich konfiguracjach, w których co najmniej jedna akcja jest przygotowana w serwerze s.

Formuły identyfikujące zakleszczenie komunikacyjne i bezczynność serwera podano w tabeli 3.

Tabela 3. Formuły czasowe do znajdowania różnych sytuacji w procesach. Źródło: [17].

Lp.	**Właściwość**	**Wzór CTL**	**Znaczenie**
1.	*2.*	*3.*	*4.*
1.	Zakleszczenie komunikacyjne w serwerze s	$\mathbf{EF\ AG}\ (D_s \wedge \neg E_s)$	W końcu zawsze $(D_s \wedge \neg E_s)$
2.	Serwer s jest bezczynny	$\mathbf{AF\ AG}\ (\neg D_s)$	W każdej realizacji ostatecznie zawsze $(\neg D_s)$
3.	Zakleszczenie na zasobach agenta a	$\mathbf{EF\ AG}\ (D_a \wedge \neg E_a)$	W końcu zawsze $(D_a \wedge \neg E_a)$
4.	Terminacja agenta a	$\mathbf{AF}(F_a)$	W każdej realizacji ostatecznie (F_a)

Zakleszczenie na zasobach definiuje się jako sytuacje, w których komunikat procesu agenta oczekuje na serwerze, ale nigdy nie będzie odpowiadał żadnemu stanowi tego serwera. Terminacja agenta to po prostu wykonanie jego akcji zakończenia. Wymagane są następujące oznaczenia LTS:

- D_a = prawda, we wszystkich konfiguracjach, w których wiadomość agenta a jest oczekująca,
- E_a = prawda, we wszystkich konfiguracjach, w których akcja z wiadomością agenta a na wejściu jest przygotowana,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- F_a = prawda, we wszystkich konfiguracjach, w których przygotowano akcję terminującą (z wiadomością agenta a na wejściu).

Wzory w logice CTL (computational tree logic) na wpadnięcie agenta w zakleszczenie na zasobach oraz na zakończenie pracy agenta podano w tabeli 3.

Operator **AG** ma znaczenie "zawsze na wszystkich ścieżkach". Operator **EF** oznacza "istnieje taka ścieżka". Operator **AF** oznacza "kiedyś na każdej ścieżce".

Znalezienie zakleszczenia dotyczy poszczególnych procesów, a nie całego systemu. Ta cecha odróżnia prezentowaną technikę od innych statycznych podejść do wykrywania zakleszczeń poprzez weryfikację modelową.

Wzory na wykrywanie zakleszczeń i sprawdzanie terminacji podane w tabeli 3 nie zależą od struktury weryfikowanego systemu. Dlatego wszystko, co musi zrobić użytkownik, to specyfikacja systemu jako zbiór akcji nad zbiorami wiadomości i stanów. Następnie weryfikacja jest przeprowadzana automatycznie w Dedanie w stylu "naciśnij przycisk", podając kontrprzykłady w przypadku znalezienia zakleszczenia.

2.8 Notacja graficzna DA^3

Weryfikacja układu (systemu) poddawanego modelowaniu przebiega w przestrzeni osiągalności, utworzonej z konfiguracji początkowej poprzez zastosowanie zdefiniowanych akcji. Projektowanie modelu najlepiej wykonać w wersji graficznej systemu DA^3, która jest formalnie zgodna z IMDS, ale jest bliższa intuicji projektanta.

Formalizm DA^3 (ang. Distributed Autonomous and Asynchronous Automata, in. DAAA) odzwierciedla zachowanie rozproszonych komponentów [18].

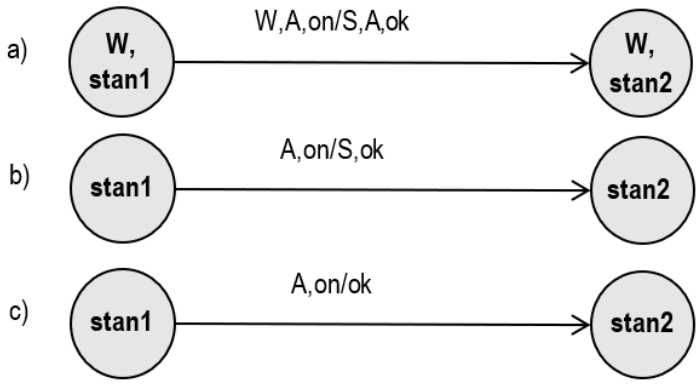
Weryfikowany system może zostać przedstawiony jako model będący złożeniem automatów rozproszonych DA^3, w których każdy automat reprezentuje zmiany stanów konkretnego serwera na skutek wykonywania akcji wyzwalanych komunikatami agentów (widok serwerów [18]). Akcja IMDS w postaci pary par ((*komunikat, stan*), (*nowy komunikat, nowy stan*)) (w rozdziale 2.3 opisana worem (10): $\lambda = \big((m,p),(m',p')\big)$) jest opisywana przez łuk (tranzycję) pomiędzy wierzchołkami grafu w automacie prowadzącą ze stanu do nowego stanu, z etykietą będącą parą (*komunikat, nowy komunikat*) [35].

Do celu projektowania, w którym używane są typy serwerów i ich instancje, dla stanów jest używana notacja (*serwer, wartość*), gdzie wartości odróżniają różne stany tego samego serwera

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

(notacja ta przedstawiona jest za pomocą pary (s, v) w języku programowania opisanym w rozdziale 2.4). Ten sam agent może wywoływać na tym samym serwerze różne akcje. Aby to zamodelować, wprowadza się w serwerach pojęcie usługi.

Usługi rozróżniają akcje serwera. Komunikat jest wywołaniem przez agenta określonej usługi na danym serwerze. Dla komunikatów jest w związku z tym używana notacja (*serwer, agent, usługa*) (notacja ta przedstawiona jest za pomocą trójki (a, s, r) w języku programowania opisanym w rozdziale 2.4). Pełna etykieta akcji w automacie ma więc postać (*serwer, agent, usługa*)/(*docelowy serwer, agent, docelowa usługa*). Ponieważ serwer jest znany (jest to nazwa projektowanego automatu), a agent w obu komunikatach jest ten sam, to w uproszczonej wersji akcja ma postać (*agent, usługa*)/(*docelowy serwer, docelowa usługa*). Dodatkowo, jeżeli docelowy serwer jest oczywisty - agent może pojawić się tylko przysyłając komunikat z konkretnego sewera – i jest to tylko odpowiedź potwierdzająca wykonanie akcji, to docelowy serwer może zostać pominięty. Te trzy przypadki zostały przedstawione na rysunku 7. W etykietach stanów i łuków pominięto nawiasy. Na rysunku 7, wykorzystując oznaczenia wprowadzone w rozdziale 2.4, można zapisać, że: $a = A, s = W, s' = S, r = on, r' = ok, v = stan\ 1, v' = stan\ 2.$



Rysunek 7. Pełny opis akcji jako łuk w grafie automatu, oraz jego uproszczenia. a – pełna akcja, b – uproszczona etykieta, c – etykieta bez serwera docelowego. Źródło: opracowanie własne na podstawie [35]

Jest to akcja w serwerze *W*, więc jego nazwa może zostać pominięta w stanach (*W,stan1*) i (*W,stan2*). W akcji agent *A* wywołuje usługę *on* na serwerze *W*. Akcja ta powoduje wysłanie komunikatu agenta do serwera *S* wywołując jego usługę *ok*. Stany *stan1* i *stan2* odnoszą się do tego samego serwera, i jest to ten sam serwer, do którego jest skierowany komunikat wejściowy, w tym przypadku *W*. Stany będące parami jak (*W, stan1*) można zatem zastąpić

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

samymi wartościami stanów jak *stan1*. Można też pominąć serwer w komunikacie wejściowym (*serwer, agent, usługa*) pozostawiając parę (*agent, usługa*), na przykład (*A*, *on*). Komunikat wyjściowy dotyczy tego samego agenta co wejściowy, więc można pominąć agenta w komunikacie wyjściowym co skraca go do pary (*serwer, usługa*), jak (*S,ok*). Nie można natomiast w ogólnym przypadku pominąć serwera, do którego jest skierowany komunikat wyjściowy. W wyjątkowych przypadkach, gdy serwer w komunikacie wyjściowym jest znany, można pozostawić tylko usługę. Sytuacja taka dotyczy jednak tylko konkretnych przypadków, w których "naturalne" jest skierowanie komunikatu do określonego serwera, na przykład skierowanie odpowiedzi *ok* do serwera (np. *S*), który jest jedynym źródłem komunikatu wejściowego (dla serwera W, dla którego narysowano wierzchołki i łuki w grafie na rysunku 7).

2.9 Weryfikacja w formalizmie IMDS

Weryfikacja w formalizmie IMDS polega na przejrzeniu grafu osiągalności konfiguracji. Podstawowy algorytm weryfikacji w logice temporalnej CTL polega [5] na znalezieniu punktu stałego pewnego funkcjonału. Algorytm weryfikacji działa tak, jakby przeglądał wszystkie możliwe sekwencje wykonania systemu jednocześnie (a nie jedna po drugiej, bo jest to niewykonalne), poszukując konfiguracji o określonej charakterystyce. Weryfikator Dedan używa algorytmu CBS [14] do znajdowania następujących własności systemu: zakleszczenie, nieuchronna terminacja i możliwa terminacja. W tym celu, dla każdej konfiguracji T w grafie osiągalności jest określana wartość logiczna zdań:

- $D_T(s), s \in S$ — zdanie prawdziwe dla każdego serwera s, dla którego w konfiguracji T występuje komunikat jakiegoś agenta a skierowany do serwera s, czyli $\exists_{p \in P \wedge p=(a,s',r)} p \in T \wedge s = s'$,
- $E_T(s), s \in S$ — zdanie prawdziwe dla każdego serwera s dla którego w konfiguracji T co najmniej jedna akcja jest umożliwiona na węźle s, czyli $\exists_{\lambda \in \Lambda \wedge \lambda=\left(\left((a,s',r),(s',v)\right),\left((a,s'',r''),(s',v')\right)\right)} (a,s',r) \in T \wedge (s',v) \in T \wedge s = s'$.

Zakleszczenie serwera s definiuje się jako konfigurację, w której w węźle oczekuje co najmniej jeden komunikat, ale w teraźniejszości i przyszłości tego serwera (w grafie osiągalności w konfiguracji T i wszystkich konfiguracjach osiągalnych z T) nie jest umożliwiona żadna akcja na węźle s. Jest to zapisane zdaniem temporalnym w logice CTL: $\mathbf{AG}\left(D_T(s) \wedge \neg E_T(s)\right)$. Operator **AG** ma znaczenie "zawsze na wszystkich ścieżkach", oczywiście zaczynających się

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

w konfiguracji T. W systemie zawsze jest stawiane pytanie o możliwość zakleszczenia, więc pełna formuła przyjmuje postać **EF AG** $\left(D_T(s) \wedge \neg E_T(s)\right)$. Operator **EF** oznacza "istnieje taka ścieżka". Całe zdanie można więc odczytać jako "na każdej ścieżce istnieje taka konfiguracja, w której w węźle oczekuje komunikat, ale w teraźniejszości T i przyszłości tego serwera nie może zostać wykonana żadna akcja". Ponieważ prawdziwość zdań $D_T(s)$ i $E_T(s)$ może być określona statycznie, to można automatycznie zbadać występowanie zakleszczeń danego serwera bez znajomości formuł temporalnych, są one ukryte w weryfikatorze Dedan. Można również zapytać o wpadanie w zakleszczenie zbioru serwerów, wtedy formuła będzie miała postać $S_x = \{s_1, s_2, \dots\} \subset S$:**EF AG** $(D_T(s_1) \wedge \neg E_T(s_1) \wedge D_T(s_2) \wedge \neg E_T(s_2) \wedge \dots)$. Formuła taka jest również automatycznie weryfikowana przez program Dedan po wskazaniu zbioru serwerów (w szczególności mogą to być wszystkie serwery).

Dla każdej konfiguracji T jest określona również prawdziwość zdań logicznych:

- $D_T(a), a \in A$ — zdanie prawdziwe dla każdego agenta a dla którego w konfiguracji T występuje komunikat agenta a, czyli $\exists_{p \in P \wedge p=(a\prime,s,r)} p \in T \wedge a = a'$, czyli agent nie uległ terminacji.
- $E_T(a), a \in A$ — zdanie prawdziwe dla każdego agenta a dla którego w konfiguracji T co najmniej jedna akcja jest umożliwiona z udziałem agenta a, czyli $\exists_{\lambda \in \Lambda \wedge \lambda = \left((a\prime,s,r),(s,v)\right),\left((a\prime,s\prime,r\prime),(s,v')\right)} (a', s', r) \in T \wedge (s, v) \in T \wedge a = a'$.
- $F_R(a), a \in A$ — zdanie prawdziwe dla każdego agenta a dla którego w konfiguracji T nie występuje jego komunikat, czyli $\forall_{m \in T, m=(a',s,r)} a \neq a'$.

Zakleszczenie agenta a definiuje się jako konfigurację, w której występuje komunikat agenta a, ale w teraźniejszości T i przyszłości tego agenta (w grafie osiągalności w konfiguracji T i wszystkich konfiguracjach osiągalnych z T) nie jest umożliwiona żadna akcja mająca na wejściu komunikat agenta a. Ponieważ stawiane jest pytanie o możliwość zakleszczenia, jest to zapisane zdaniem temporalnym w logice CTL: **EF AG** $\left(D_T(a) \wedge \neg E_T(a)\right)$. Analogicznie do serwerów, można spytać o wspólne zakleszczenia zbioru agentów, wtedy formuła będzie miała postać $A_x = \{a_1, a_2, \dots\} \subset A$:**EF AG** $(D_T(a_1) \wedge \neg E_T(a_1) \wedge D_T(a_2) \wedge \neg E_T(a_2) \wedge \dots)$, a nawet wszystkich agentów. Należy podkreślić, że algorytm TempoRG działa na zasadzie silnej sprawiedliwości [15], co oznacza że dla dowolnego cyklu w grafie osiągalności, jeżeli jest możliwe opuszczenie tego cyklu, to po wielu możliwych wykonaniach cyklu musi on zostać

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

opuszczony. Ta właściwość gwarantuje, że system nie utknie w żadnym "zagłodzeniu" jego któregoś z serwerów, jeżeli może on wykonać określoną akcję.

Terminacja to sytuacja, w której agent wykonał akcję terminującą, a więc nieuchronność terminacji to występowanie na każdej ścieżce w grafie osiągalności konfiguracji nie zawierającej komunikatu agenta, co jest wyrażone formułą **AF** $\left(F_T(a)\right)$. Operator **AF** oznacza "kiedyś na każdej ścieżce".

Możliwa terminacja z kolei jest sytuacją, w której konfiguracja nie zawierająca komunikatu agenta występuje na jakiejś ścieżce, co wyraża formuła **EF** $\left(F_T(a)\right)$.

Weryfikacja polega więc na wskazaniu serwerów/agentów i cechy, którą chcemy sprawdzić. Na przykład nieuchronna terminacja agentów, którzy ulegają terminacji po zaobserwowaniu określonych pożądanych zdarzeń w systemie.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

3 Uogólniony model interfejsu przekaźnikowego urządzeń komputerowych

3.1 Wprowadzenie

Niniejszy rozdział został opracowany na podstawie doświadczeń własnych autora rozprawy obejmujących badania przekaźników, komputerowych systemów srk i ich interfejsów. Zastosowane opisy i oznaczenia mają charakter ogólny, pozwalający zastosować je do wielu różnych istniejących lub projektowanych interfejsów przekaźnikowych urządzeń komputerowych srk. Wykorzystanie wektorów do opisu sygnałów wejściowych i wyjściowych systemu srk, przesyłanych z i do otoczenia poprzez interfejs, zostało opisane m.in. w pracy [80].

Przedstawione treści stanowią wcześniejszą próbę podejścia autora rozprawy do problemu opisu i badania interfejsów i powiązań. Treści te nie znalazły zastosowania w opracowanej ostatecznie w rozprawie metodzie ponieważ zastosowano opis w języku IMDS i notacji DA^3 niezbędny do badania modeli w środowisku Dedan. Autor rozprawy uznał ostatecznie, że ścieżka takiego opisu nie doprowadzi go do realizacji celu rozprawy. Może to stanowić istotną wskazówkę dla badaczy, którzy chcieliby tę metodę rozwijać. Tym niemniej zaprezentowane treści mogą stanowić element opisu przydatny w innych badaniach z zakresu tematyki.

3.2 Wyjścia

Wyjścia interfejsu urządzeń komputerowych, za pomocą których urządzenia te są w stanie przesyłać do współpracujących z nimi urządzeń sygnały, najczęściej są bezpotencjałowe i obejmują układy złożone z zestyków przekaźników znajdujących się w strukturze systemu komputerowego. Najczęściej wmontowane w karty obsługujące wyjścia.

Wyjścia takie mogą być różnych rodzajów pod względem zasady działania i konstrukcji fizycznej ale logicznie realizują funkcję zestyku łączącego lub rozłączającego na polecenie urządzeń zależnościowych dwa wyprowadzenia danego wyjścia.

Zasada działania może polegać na tym, że wyjście takie jest normalnie zwarte lub normalnie rozwarte.

Konstrukcja fizyczna może obejmować wykorzystanie przekaźników elektromagnetycznych lub elementów elektronicznych (tranzystorów, triaków, tyrystorów).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Układy zestyków określonego wyjścia mogą obejmować zestyki zwierne i rozwierne różnych przekaźników łączone szeregowo i/lub równolegle w zależności od sposobu sterowania cewek.

Wektor $\mathbf{UZ\text{-}P}(\boldsymbol{Sk})$ jest wektorem reprezentującym sygnały sterujące, poprzez które procesor steruje cewkami grupy przekaźników, należących do zbioru $P(Sk)$, związanej z realizacją Sk-tej funkcji logicznej. Funkcji logicznych jest K.

k – numer funkcji logicznej, $k \in \{1,2,\dots,K\}$

$P_{Sk} = \{C_{p1,Sk}, C_{p2,Sk}, \dots, C_{pn,Sk}, \dots C_{pN,Sk}\}$ – zbiór przekaźników grupy realizującej Sk-tą funkcję logiczną

$C_{pn,Sk}$ – n-ty przekaźnik grupy realizującej Sk-tą funkcję logiczną

Wektor $\mathbf{UZ\text{-}P_{Sk}}$ może mieć składowe reprezentujące prąd lub napięcie doprowadzone do danej cewki, co umożliwić ma wzbudzenie w niej strumienia magnetycznego o wartości potrzebnej do zmiany stanu danego przekaźnika.

Takie podejście powoduje, że dla zbioru przekaźników powinny zostać określone funkcje $wo_{Sk}(\mathbf{UZ\text{-}P_{Sk}})$ określające relację pomiędzy wartością wielkości fizycznej sygnału sterującego (prądu lub napięcia) a zmianą stanu danego przekaźnika. Jest to zgodne z rzeczywistym działaniem przekaźnika. Zmiennymi takich funkcji będą przede wszystkim spadek napięcia lub natężenie prądu. Będą to funkcje nieliniowe, opierające się na histerezie, co wynika z zasady działania przekaźnika elektromagnetycznego. Kolejne zmienne mogą reprezentować zużycie przekaźnika, wpływ temperatury otoczenia itp.

Jeżeli jedna ze składowych wektora $\mathbf{UZ\text{-}P_{Sk}}$ zostanie oznaczona jako natężenie prądu przepływające przez cewkę pierwszego przekaźnika $i_1(t)$ wówczas funkcja $wo_{Sk}(i_1(t))$ dla tego przekaźnika, uwzględniająca histerezę może zostać sformułowana w następujący sposób:

$$wo_{Sk}(i_1(t)) = \begin{cases} 0 \Leftrightarrow \begin{cases} (\dot{\imath}(t) > 0) \wedge (i(t) < i_{1,on}) \\ (\dot{\imath}(t) < 0) \wedge (i(t) < i_{1,off}) \end{cases} \\ 1 \Leftrightarrow \begin{cases} (\dot{\imath}(t) > 0) \wedge (i(t) > i_{1,on}) \\ (\dot{\imath}(t) < 0) \wedge (i(t) > i_{1,off}) \end{cases} \end{cases} \tag{27}$$

Gdzie:

$i_1(t)$ – wartość natężenia prądu przepływającego przez cewkę pierwszego przekaźnika,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

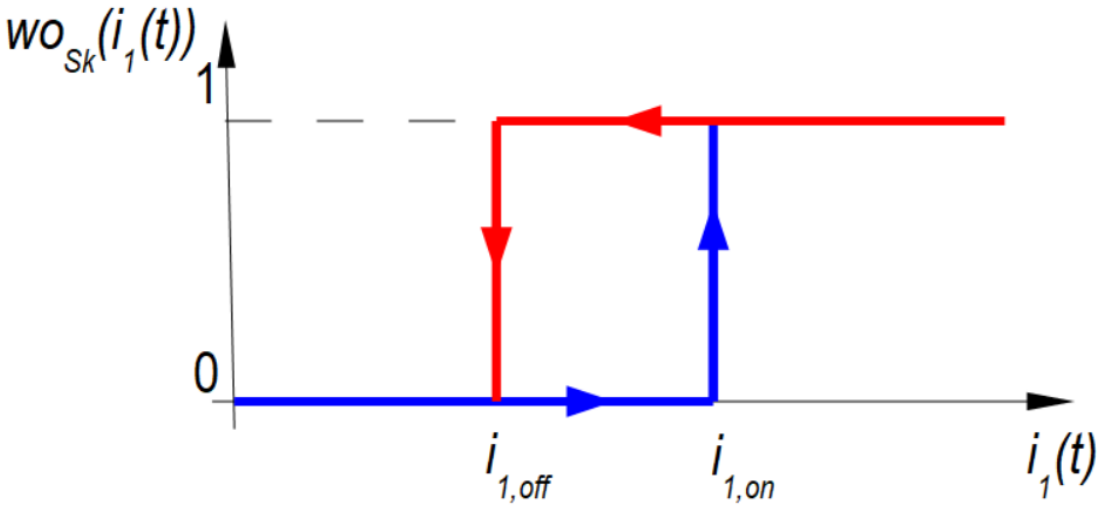
$i_{1,on}$ – wartość natężenia prądu przepływającego przez cewkę pierwszego przekaźnika, przy której przekaźnik przejdzie do stanu czynnego (wzbudzi się),

$i_{1,off}$ – wartość natężenia prądu przepływającego przez cewkę pierwszego przekaźnika, przy której przekaźnik przejdzie do stanu biernego (odwzbudzi się),

$wo_{Sk}(i_1(t))$ =0 oznacza stan bierny przekaźnika (przekaźnik jest odwzbudzony),

$wo_{Sk}(i_1(t))$ =1 oznacza stan czynny przekaźnika (przekaźnik jest wzbudzony).

Powyższe równanie ilustruje również wykres na rysunku 8.



Rysunek 8. Wykres charakterystyki prądowej przekaźnika uwzględniający zjawisko histerezy. Źródło: opracowanie własne

Powyższa charakterystyka nie uwzględnia ograniczenia jakie wynika z możliwości działania przekaźnika przy prądzie płynącym w przeciwnym kierunku ani przekroczenia przez określony czas wartości natężenia prądu, przy której nastąpi trwałe uszkodzenie cewki: przerwa (przepalenie drutu nawojowego) lub zwarcie (przebicie izolacji elektrycznej pomiędzy uzwojeniami cewki).

W sytuacji, w której wpływ napięcia lub prądu nie będzie interesujący z punktu widzenia badań, składowe wektora **UZ-P$_{Sk}$** mogą być reprezentowane poprzez zmienne boolowskie:

- prawdę (1) – reprezentowaną poprzez wysyłanie sygnału mającego wzbudzić dany przekaźnik,
- fałsz (0) – reprezentowany poprzez wysyłanie sygnału mającego dany przekaźnik odwzbudzić.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

W obu przypadkach (zmiennych reprezentujących prąd lub napięcie albo boolowskich) funkcje $wo_{Sk}(\mathbf{UZ\text{-}P_{Sk}})$ powinny zależeć od czasu t, ponieważ przekaźnik elektromagnetyczny zmienia swój stan w określonym czasie zależnym od chwili zmiany wartości napięcia (lub prądu) podanego na cewkę. Ponadto w trakcie tej zmiany występują stany pośrednie, konieczne do uwzględnienia w modelu ze względu na konstrukcję przekaźnika oraz wykorzystywanie tej cechy w obwodach srk.

Jeżeli rozważone zostanie zasilanie napięciem znamionowym cewki przekaźnika, prąd narasta według określonej charakterystyki. Po osiągnięciu określonej wartości (wcześniej dla przekaźnika 1 opisanej jako $i_{1,on}$) kotwica zostaje wprawiona w ruch. Ruch ten wpływa asynchronicznie na położenie zestyków (zestyki nie przełączają się w tym procesie w jednej chwili czasu). Proces przełączania jest zależny od ogólnie sformułowanych dla przekaźnika czasów określających:

t_{rr} – czas upływający od podania napięcia znamionowego na cewkę do chwili rozwarcia zestyków rozwiernych,

t_{zz} – czas upływający od podania napięcia znamionowego na cewkę do chwili zwarcia zestyków zwiernych.

Co do zasady, jeśli nie zastosowano specjalnej konstrukcji zestyku (tzw. zestyku przełącznego pod prądem), $t_{rr} < t_{zz}$.

Jeżeli zasilanie cewki przekaźnika zostanie przerwane, wartość prądu spada do zera, a kotwica rozpoczyna ruch powrotny. Jeśli wartość prądu jest regulowana (np. poprzez zmianę spadku napięcia na cewce), wówczas skutek taki powoduje obniżenie natężenia poniżej określonej wartości (wcześniej dla przekaźnika 1 opisanej jako $i_{1,off}$). Ruch ten wpływa asynchronicznie na położenie zestyków. Proces przełączania jest zależny od ogólnie sformułowanych dla przekaźnika czasów określających:

t_{rz} – czas upływający od zaprzestania zasilania cewki do chwili rozwarcia zestyków zwiernych,

t_{zr} – czas upływający od zaprzestania zasilania cewki do chwili zwarcia zestyków rozwiernych.

Co do zasady, jeśli nie zastosowano specjalnej konstrukcji zestyku (tzw. zestyku przełącznego pod prądem), $t_{rz} < t_{zr}$.

Funkcje $wo_{Sk}(\mathbf{UZ\text{-}P_{Sk}})$ będą wiązać działanie cewki z układem zestyków. Dla uproszczenia i standardowego przekaźnika można wprowadzić jedną funkcję określająca wpływ na zestyk

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

zwierny i jedną na zestyk rozwierny – powtarzalne dla pozostałych zestyków (a w przypadku zestyków przełącznych zastępowanie ich logicznie układem połączonych zestyków zwiernego i rozwiernego).

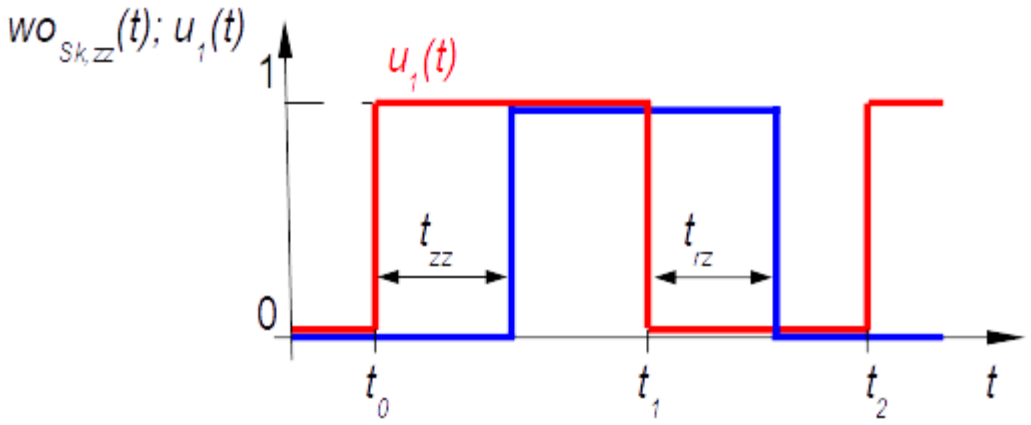
Przykładowa funkcja $wo_{Sk,zz}(t)$ uzależniająca stan zestyków zwiernych przekaźnika k-tej funkcji od obecności lub nie napięcia na cewce tego przekaźnika może mieć następującą postać:

$$\forall\big((t_0 < t_1 < t_2) \wedge (t_0, t_1, t_2 \in T) \wedge (t_1 \geq t_0 + t_{zz}; t_2 - t_1 \geq t_{rz})\big)\, wo_{Sk,zz}(t) = \begin{cases} 0 \Leftrightarrow (t < t_0) \wedge (t \geq t_1) \\ 1 \Leftrightarrow (t \geq t_0) \wedge (t < t_1) \wedge (t \geq t_2) \end{cases} \tag{28}$$

Gdzie:

T – dziedzina czasu (zbiór chwil).

Powyższą funkcję można zilustrować wykresem przedstawionym na rysunku 9.



Rysunek 9. Charakterystyka czasowa przełączania zestyków zwiernych przekaźnika. Źródło: opracowanie własne

Na rysunkach 10 i 11 przedstawiono również wektor $\mathbf{P\text{-}UZ_{Sk}}$ reprezentujący sygnały zwrotne pochodzące od cewek, które może brać pod uwagę procesor. Wektor ten oznaczono linią przerywaną ponieważ w praktyce procesor analizuje stan przekaźników inną drogą (pętla zamyka się poprzez układ zestyków kontrolowanych przekaźników).

$$wo_{Sk}\colon \mathbf{UZ\text{-}P_{Sk}} \longrightarrow \mathbf{C\text{-}Z_{Sk}}$$

Funkcje wo_{Sk} przekształcają składowe wektora $\mathbf{UZ\text{-}P_{Sk}}$ w składowe wektora $\mathbf{C\text{-}Z_{Sk}}$. Wektor $\mathbf{C\text{-}Z_{Sk}}$ o składowych będących zmiennymi boolowskimi reprezentuje sygnały logiczne pochodzące od cewek a kierowane do zestyków przekaźników. Reprezentuje to oddziaływanie mechaniczne układu przełączającego zestyki.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Reprezentacja taka nie będzie kompletna, jeśli nie weźmie się pod uwagę relacji pomiędzy zestykami. W przekaźnikach stosowanych w srk realizowana jest funkcja sztywnego prowadzenia zestyków[6] gwarantująca właściwe sekwencje przełączania zestyków określonych rodzajów. Relacje te mogą zostać ujęte w postaci funkcji wo_{Sk} poprzez właściwy dobór stałych reprezentujących czasy przełączania, ale należy pamiętać, że pozwala to wyłącznie na zamodelowanie prawidłowej sekwencji działania podczas pracy nieuszkodzonego przekaźnika (brak uszkodzeń polegających na zgrzaniu zestyku).

Ze względu na opis układu znajdującego się na wyjściu systemu komputerowego, zgrzanie takie, pomimo zastosowania środków ograniczających ryzyko w postaci kodeksu postępowania (zaprojektowanie obwodów na właściwe napięcie i prąd; zastosowanie wyłączników nadmiarowo-prądowych o odpowiedniej wartości; właściwa eksploatacja obwodów i ich utrzymanie) może mieć miejsce. Stąd funkcje logiczne $zr_{Sk}(\mathbf{C\text{-}Z_{Sk}}, \mathbf{Up_{Sk}})$. Wektor $\mathbf{Up_{Sk}}$ reprezentuje uszkodzenia poszczególnych zestyków w postaci zmiennych boolowskich:

- prawda (1) – zestyk uszkodzony,
- fałsz (0) – zestyk sprawny.

Zestyki oznaczono indeksami dolnymi dodatkowymi względem oznaczeń dla cewek – pierwsza zmienna oznacza rodzaj i jednocześnie numer zestyku danego rodzaju:

- r – zestyk rozwierny,
- z – zestyk zwierny.

$z_{rpn,pn,Sk}$ – zestyk rozwierny numer rpn w przekaźniku pn-tym grupy realizującej Sk-tą funkcję logiczną

$$rpn \in \{1, 2, \dots, RPN\}$$

$z_{zpn,pn,Sk}$ – zestyk zwierny numer zpn w przekaźniku pn-tym grupy realizującej Sk-tą funkcję logiczną

$$zpn \in \{1, 2, \dots, ZPN\}$$

Wartości RPN i ZPN są cechami konstrukcji danego pn-tego przekaźnika.

[6] Zgodnie z fiszką UIC-736 [61] oraz normami IEC [54][56] funkcja ta polega na tym, że konstrukcja przekaźnika uniemożliwia jednoczesne pozostawanie zamkniętym któregokolwiek z grupy jego zestyków zwiernych z którymkolwiek z grupy jego zestyków rozwiernych.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

W praktyce w układach interfejsów systemów komputerowych stosuje się jednakowe karty zawierające zazwyczaj ten sam typ i odmianę przekaźnika. Oznacza to w modelu przyjęcie wspólnej wartości RPN dla wszystkich przekaźników powiązania i wspólnej wartości ZPN dla wszystkich przekaźników powiązania.

Należy zwrócić uwagę na brak wektora o zwrocie przeciwnym do $\mathbf{C\text{-}Z_{Sk}}$. Wektor taki reprezentowałby oddziaływanie zestyków na cewkę, co praktycznie nie występuje. Można byłoby rozważać takie oddziaływanie, gdyby był potrzebny wektor $\mathbf{P\text{-}UZ_{Sk}}$ reprezentujący składowymi wartości indukcyjności cewek i występowałaby duża czułość mikroprocesora na zmiany jego składowych. Tymczasem, na potrzeby modelu przyjmuje się wręcz uproszczenie polegające na przyjęciu boolowskich zmiennych jako składowych wektora $\mathbf{UZ\text{-}P_{Sk}}$.

Zestyki przekaźników mogą być łączone w układy realizując tym sposobem funkcje logiczne. Z reguły są to układy kombinacyjne. W przypadku dwóch przekaźników biorących udział w realizacji funkcji na wyjściu systemu komputerowego częstym rozwiązaniem jest połączenie ich zestyków w taki sposób, żeby:

- do wyjścia mającego realizować funkcję zestyku zwiernego (NO) wykorzystać połączenie szeregowe zestyku zwiernego jednego przekaźnika z zestykiem zwiernym drugiego,
- do wyjścia mającego realizować funkcję zestyku rozwiernego (NC) wykorzystać połączenie równoległe zestyku rozwiernego jednego przekaźnika z zestykiem rozwiernym drugiego.

Możliwe są również inne połączenia szeregowe, równoległe i szeregowo-równoległe. Wykonuje się je w zależności od:

- funkcji jakie ma spełniać dane wyjście,
- wymagań obwodów zewnętrznych względem systemu komputerowego, do których ma być typowo wykorzystywane,
- zasad sterowania przekaźników interfejsu przez układy zależnościowe.

Układy, w które połączone zostaną zestyki, są reprezentowane na rysunkach 10 i 11 poprzez bloki $UKZw_{Sk}$ i $UKZ_{w,Sk}$ realizujące funkcje logiczne $kz_{Sk}(\mathbf{Z\text{-}UKZ_{Sk}})$ i $cz_{w,Sk}(\mathbf{Z\text{-}UKZ_{Sk}})$ na podstawie wektora zmiennych boolowskich $\mathbf{Z\text{-}UKZ_{Sk}}$. Wektor ten ma składowe reprezentujące stan poszczególnych zestyków przekaźników ze zbioru P_{Sk}, przy czym stan danego zestyku jest reprezentowany poprzez:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- prawdę (1) – dla zestyku w stanie zwartym w danej chwili czasu,
- fałsz (0) – dla zestyku w stanie rozwartym w danej chwili czasu.

kz_{Sk}: **Z-UKZ**$_{Sk}$ ⟶ **UKZw-UZ**$_{Sk}$ – funkcja kontroli zwrotnej.

$cz_{w,Sk}$: **Z-UKZ**$_{Sk}$ ⟶ $\mathbf{w}_{w,Sk}$ – funkcja kombinacji zestyków w-tego wyjścia realizującego Sk-tą funkcję logiczną. $w \in \{1,2,\dots,W_{Sk}\}$, W_{Sk} – liczba wyjść Sk-tej funkcji logicznej.

$\mathbf{w}_{w,Sk}$ – wektor określający stan w-tego wyjścia realizującego Sk-tą funkcję logiczną.

Dana funkcja Sk obejmuje również wyjścia zanegowane, których numery są objęte wspólnym zbiorem z nie zanegowanymi.

Składowe wektora $\mathbf{w}_{w,Sk}$ reprezentować będą określone właściwości fizyczne zestyku zwartego lub rozwartego (np. opór elektryczny wyjścia zwartego i rozwartego). Dla potrzeb modelowania wystarczające jest przekształcenie w wartość 0 lub 1, co odpowiada interpretacji przedstawionej wcześniej dla zestyku. Wektor $\mathbf{w}_{w,Sk}$ może być zatem binarnym skalarem $w_{w,Sk}$, wartość którego ma taką sama interpretację, jak podana wyżej dla zestyków.

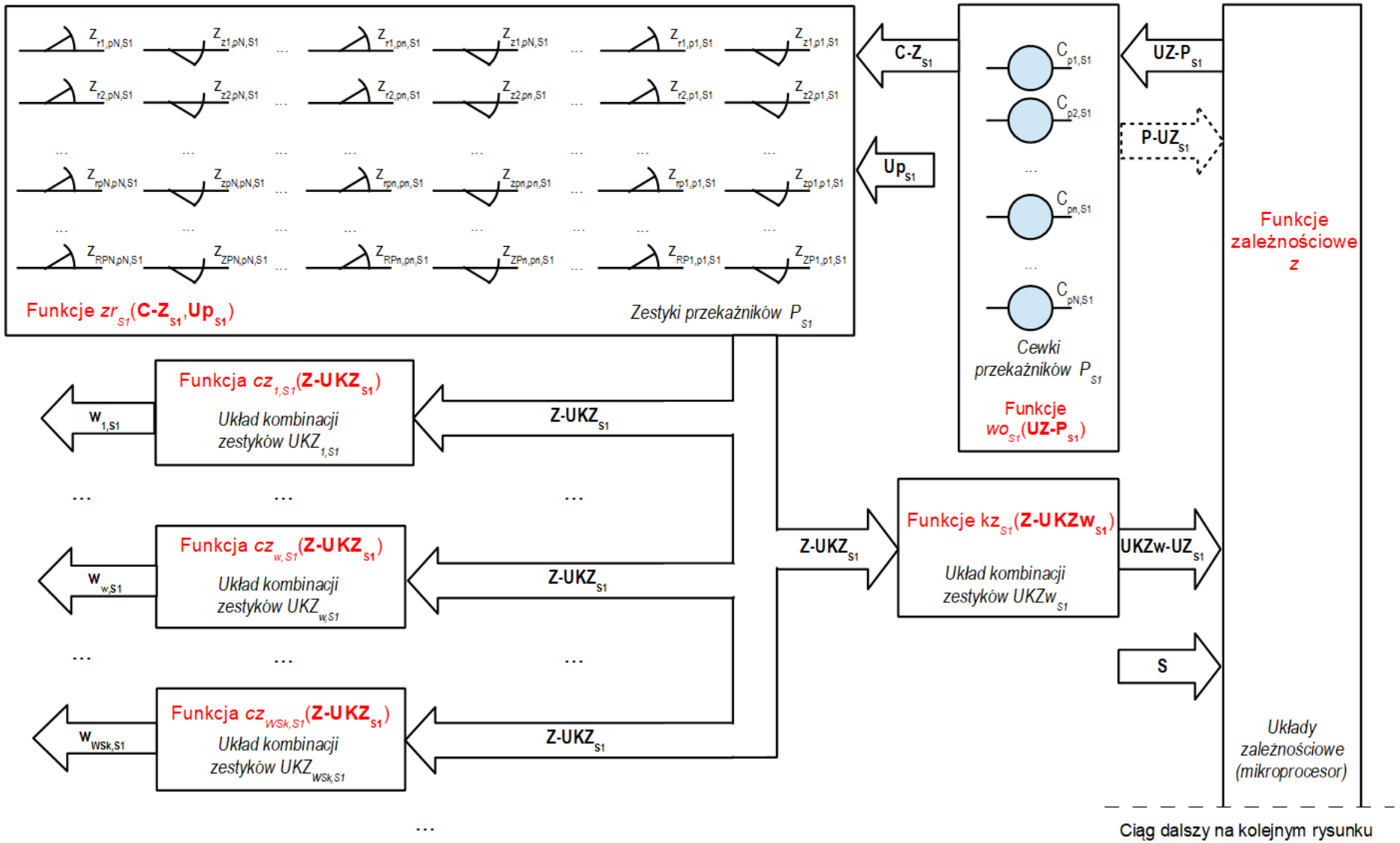
Wektor **S** reprezentuje pozostałe sygnały napływające do układów zależnościowych.

Funkcje zależnościowe można określić jako

$$z\text{:}\mathbf{S},\{\mathbf{UKZw\text{-}UZ_{Sk}}\}_{\forall k\in\{1,2,\dots,K\}} \longrightarrow \{\mathbf{UZ\text{-}P_{Sk}}\}_{\forall k\in\{1,2,\dots,K\}}$$

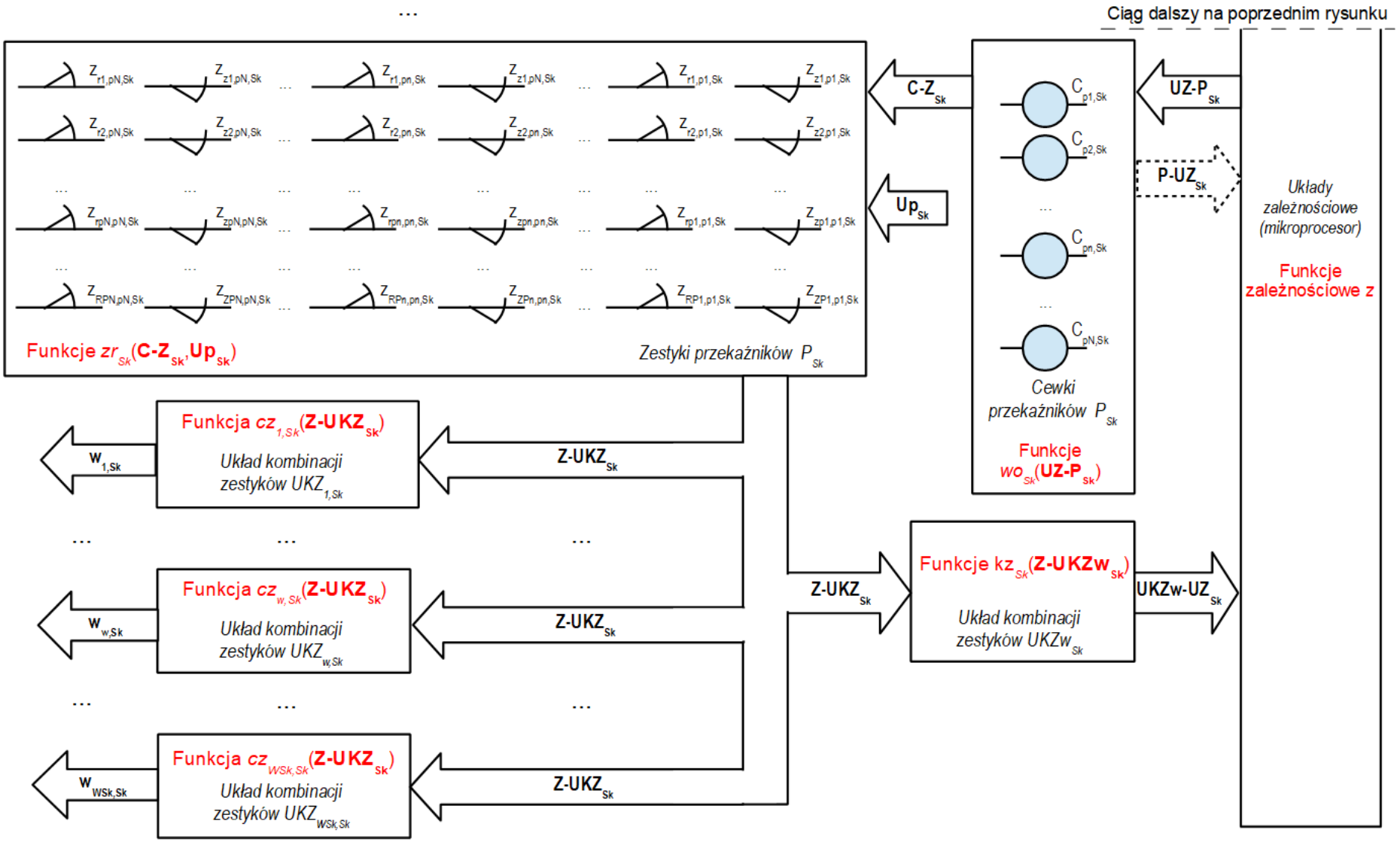
Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology



Rysunek 10. Uogólniony model interfejsu przekaźnikowego w urządzeniach komputerowych w zakresie wyjść. Część 1 z 2. Źródło: opracowanie własne

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology



Rysunek 11. Uogólniony model interfejsu przekaźnikowego w urządzeniach komputerowych w zakresie wyjść. Część 2 z 2. Źródło: opracowanie własne

3.3 Wejścia

Wejścia interfejsu urządzeń komputerowych, za pomocą których urządzenia te są w stanie odbierać od współpracujących z nimi urządzeń sygnały, najczęściej obejmują transoptory. Ich zastosowanie ma na celu odseparowanie galwaniczne układu kontrolowanego od czułych układów cyfrowych operujących najczęściej niższymi napięciami i prądami. Doprowadzenie do tych układów sygnałów wejściowych bezpośrednio wymagałoby obniżenia wartości tych parametrów elektrycznych, a w przypadku zakłóceń (przepięć, zwarć z obcymi źródłami energii elektrycznej) powodowałoby uszkodzenie tych układów.

Odczyt stanu zestyków układów przekaźnikowych (pośredniczących w powiązaniu bądź stanowiących fragment powiązanego urządzenia) wymaga doprowadzenia do opisywanego wejścia sygnału, który najczęściej jest:

- obecnością napięcia - jedynka logiczna – albo
- jego brakiem (zero logiczne).

Jeśli przekazywana informacja jest krytyczna dla bezpieczeństwa (jej przekłamanie może doprowadzić do sytuacji niebezpiecznej), wówczas stosuje się dodatkowe zabiegi mające na celu wyeliminowanie wpływu na wejście obcych sygnałów, które mogłyby pojawić się wskutek usterki. Stosowane jest np. podawanie jako jedynki logicznej sygnału przemiennego w postaci fali prostokątnej. Taka postać pozwala wykryć usterkę polegającą na zwarciu do jednego z biegunów zasilania, ponieważ urządzenia komputerowe akceptują wówczas tylko sygnał przemienny. Rozwinięcie tej zasady polega na porównaniu postaci sygnału taktującego (fali), który generuje komputer, z sygnałem otrzymanym z danego jego wejścia.

Wielokanałowe przetwarzanie zależności w systemie komputerowych urządzeń srk wymaga podania danego sygnału na wejścia każdego z kanałów. Powoduje to zwielokrotnienie wejść. Zastosowane zabiegi mogą wymagać podawania sygnałów (innych fal prostokątnych dla różnych kanałów) poprzez różne zestyki tego samego przekaźnika. Przy braku wolnych zestyków stosowany jest zabieg polegający na podawaniu tej samej fali prostokątnej (lub jej braku) z jednego zestyku za pośrednictwem diod prostowniczych na wiele wejść (w liczbie odpowiadającej liczbie kanałów).

Pewność odbioru informacji o stanie przekaźnika osiągnąć można poprzez kontrolowanie stanu jego zestyku zwiernego i rozwiernego. Zastosowanie kontroli wyłącznie jednego zestyku, może być niewystarczające, ponieważ przy usterce polegającej na przerwie w obwodzie

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

kontrolowanego zestyku jego stan będzie wciąż taki, jakiemu przypisano brak sygnału na wejściu do urządzeń komputerowych. Zastosowane przekaźniki muszą mieć ponadto sztywne prowadzenie zestyków, co jest ogólną cechą przekaźników stosowanych w urządzeniach srk. Cecha ta polega na tym, że w żadnej parze zestyków zwiernego i rozwiernego danego przekaźnika nie może być jednocześnie zwarcie obu tych zestyków.

Zastosowanie do kontroli stanu pracy przekaźnika jego zestyków może powodować konieczność zastosowania powtarzaczy[7] lub drugich przekaźników[8], jeżeli przekaźnik właściwy ma niewystarczającą liczba wolnych zestyków, które można zastosować w powiązaniu.

Zastosowanie zestyku przekaźnika jako źródła informacji o stanie przekaźnika pociąga za sobą opóźnienie w odbiorze informacji przez urządzenia komputerowe. Powodem jest czas przekazania sygnału od momentu pojawienia się napięcia na wejściu do przetworzenia go w sygnał logiczny. Czas ten uwzględnia czas zmiany fizycznej w kolejnych elementach wejścia oraz czas, który celowo musi upłynąć od momentu rozpoczęcia zmian na wejściu do chwili ustalenia się sygnału. Ma to związek ze zjawiskiem drgania zestyków przekaźnika w trakcie ich zamykania.

Liczba wejść urządzeń komputerowych zależy również od tego ile przekaźników i jak z układu pośredniczącego lub urządzeń przekaźnikowych, przekazuje informacje. Wytypowanie tych przekaźników powinno być dokonane na podstawie analizy tego, który zbiór przekaźników jest w stanie reprezentować jednoznacznie stan urządzenia, w skład którego wchodzą, biorąc pod uwagę potencjalne usterki i to, czy algorytm systemu komputerowego uwzględnia odczyt sekwencji stanów, czy samych stanów wejść. Typowym rozwiązaniem jest większa liczba przekaźników niż wynika to z prostego rachunku obliczenia liczby bitów na podstawie liczby stanów kontrolowanego urządzenia. Przekaźniki stosowane w urządzeniach srk charakteryzują się asymetrią uszkodzeń – łatwiej taki przekaźnik uszkadza się w taki sposób, że nie daje się go wzbudzić, niż tak, że nie można go odwzbudzić. Stąd też do kontroli stanu urządzenia,

[7] Powtarzacz przekaźnika – przekaźnik powtarzający stan innego przekaźnika. Przekaźnik obojętny, którego cewka jest zasilana poprzez zestyk (najczęściej zwierny) przekaźnika, którego stan ma być powtarzany.

[8] Drugi przekaźnik (i kolejne) – przekaźnik obojętny, napięciowy, którego cewka jest dołączona równolegle do innego przekaźnika. Przy braku usterek oba przekaźniki (pierwszy i drugi) wzbudzają się jednocześnie i odwzbudzają jednocześnie.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

realizacji danej funkcji w określonym momencie, wydania polecenia itd. wykorzystuje się wzbudzenie danego przekaźnika potwierdzające ten fakt.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

4 Modele elementów składowych układu przekaźnikowego

4.1 Założenia ogólne

Podstawowym zadaniem jest wyodrębnienie w układzie względnie niezależnych elementów, które współpracują ze sobą poprzez przesyłanie szeroko rozumianych sygnałów. Sygnałami tymi są:

- doprowadzenie do określonego punktu zasilania lub masy, lub
- oddziaływanie na określony zestyk.

Przy czym zupełnie bez różnicy jest, czy na zestyk oddziałuje człowiek (przycisk/przełącznik), czy też jest to zestyk sterowany przez kotwicę przekaźnika. Podstawowymi modelowanymi elementami będą zatem:

- uzwojenie,
- kotwica,
- zestyk.

Doprowadzenie do określonego punktu zasilania lub masy będzie dalej nazywane sygnałem pozytywnym (lub podaniem sygnału lub sygnałem, oznaczanym 1), a wycofanie zasilania/masy sygnałem negatywnym (brakiem sygnału, oznaczanym 0). Oddziaływanie na zestyk będzie zwarciem lub rozwarciem. Agent podający/wycofujący sygnał lub zwarcie musi wracać do elementu, z którego przybył, żeby móc propagować sygnał do dalszych elementów.

Doprowadzenie zasilania do obu wejść (zacisków cewki) uzwojenia przekaźnika (wzbudzenie strumienia magnetycznego) uruchamia agenta, który powoduje zmianę położenia kotwicy, a ta zwarcie bądź rozwarcie zestyków tego przekaźnika. Zabranie zasilania z dowolnego wejścia uzwojenia przekaźnika (odwzbudzenie strumienia magnetycznego) powoduje przeciwne działanie: również jest uruchamiany agent, powodujący zmianę położenia kotwicy i tym sposobem rozwarcie bądź zwarcie zestyków.

Uzwojenie działa jak bramka logiczna AND [72], ponieważ wymaga podania sygnałów z obu stron żeby się wzbudzić: sygnału obu biegunów zasilania (zasilanie i masa). Element kotwicy uruchamia własnego agenta sterującego dalej zestykami przekaźnika.

Przekaźniki stosowane w rzeczywistym układzie przełączają zestyki w taki sposób, że pomiędzy stanem zwarcia zestyków rozwiernych i stanem zwarcia zwiernych jest stan,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

w którym wszystkie zestyki są otwarte jednocześnie (jest to standardowa funkcja nazywana sztywnym prowadzeniem zestyków, realizowana w przekaźnikach stosowanych w układach związanych z bezpieczeństwem [52][59][58], w tym kolejowych [61][54][56], i gwarantująca możliwość rozważania ograniczonej liczby stanów, w jakich może znaleźć się przekaźnik, w tym w trakcie usterki). Ponieważ sterowane zestyki są mechaniczne, to założono, że zestyki danego rodzaju (zwierne albo rozwierne) mogą zostać zwarte bądź rozwarte w dowolnej kolejności. Dopiero po zmianie stanu zestyków danego rodzaju zmianie może ulegać drugi rodzaj:

- Przy wzbudzaniu przekaźnika stanem początkowym są otwarte zestyki zwierne i zamknięte rozwierne. Następnie otwierają się wszystkie zestyki rozwierne i dopiero potem zamykają wszystkie zwierne.
- Przy odwzbudzaniu przekaźnika stanem początkowym są zamknięte zestyki zwierne i otwarte rozwierne. Następnie otwierają się wszystkie zestyki zwierne i dopiero potem zamykają wszystkie rozwierne.

Nie zamodelowano równoczesnego zadziałania zestyków danego rodzaju, ponieważ model matematyczny jest przeplotowy, ale Manna i Pnueli udowodnili, że układ przeplotowy który działa diamentowo (najpierw pierwsza akcja a potem druga lub odwrotnie) jest równoważny co do zachowania układowi koincydencyjnemu [46].

Kotwica została zamodelowana w trzech odmianach:

- Dla sterowania zestykami zwiernymi (przekaźnik wyłącznie z zestykami zwiernymi),
- Dla sterowania zestykami rozwiernymi (przekaźnik wyłącznie z zestykami rozwiernymi),
- Dla sterowania zestykami zwiernymi i rozwiernymi (przekaźnik z zestykami zwiernymi i rozwiernymi).

W badaniach założono jednokierunkową propagację sygnałów (od źródła zasilania/masy do elementów, te do kolejnych elementów itd.). W takiej sytuacji zestyk działa jak bramka logiczna AND [72], to znaczy musi być obecny sygnał na wejściu i sygnał zwarcia mechanicznego, żeby sygnał został przesłany na wyjście. Sygnał wyjściowy – propagacja zasilania lub masy – może sterować wieloma elementami, co jest założone, że dzieje się w czasie zerowym, więc sygnał ten może być sekwencyjnie wysyłany do tych elementów, w arbitralnej kolejności. Zakłada się również, że ten agent, który doprowadza do zadziałania

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

zestyku (podający sygnał wejściowy lub powodujący zwarcie) propaguje sygnał wyjściowy na wyjście. Analogicznie dzieje się w przypadku wycofania sygnału lub rozwarcia zestyku.

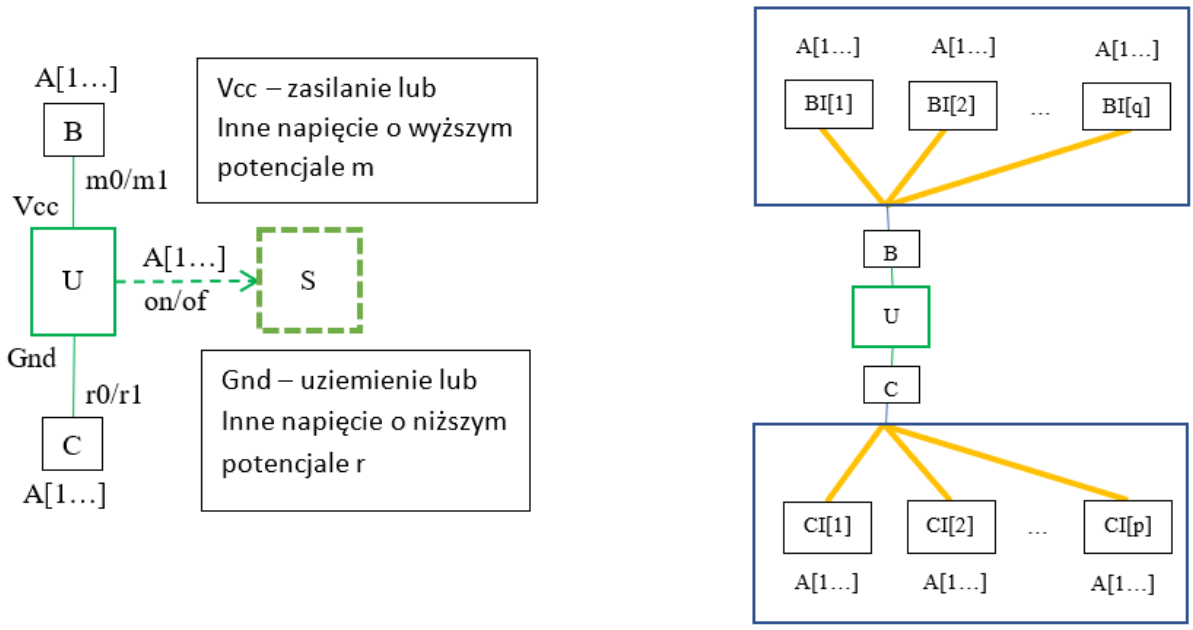
Specjalną wersją zestyku jest taki, który podaje sygnał na wyjście jako jedno z wielu źródeł. W tym przypadku ograniczono liczbę elementów na wyjściu do jednego ($j = 1$).

Ostatnim typem elementu, zawierającym dwie odmiany, jest połączenie galwaniczne:

- działające jak bramka logiczna OR [72] – podanie sygnału na którekolwiek wejście takiego połączenia powoduje podanie sygnału na wyjście, a wycofanie sygnału z wszystkich wejść powoduje wycofanie sygnału z wyjścia,
- działające jak rozgałęzienie – podanie sygnału na wejście takiego połączenia powoduje podanie sygnału na wszystkie wyjścia, a wycofanie sygnału z wejścia powoduje wycofanie sygnału z wszystkich wyjść.

4.2 Model uzwojenia

Na rysunku 12 przedstawiono schemat poglądowy modelu uzwojenia przekaźnika.



Rysunek 12. Schemat poglądowy uzwojenia przekaźnika (po lewej). Sytuacja zastosowania wielu źródeł sygnałów sterujących (po prawej). Źródło: opracowanie własne na podstawie [35]

Literą B i C oznaczono aparaty podające dwa bieguny zasilania:

- C jedno napięcie, np. masę:
  - r1 – komunikat oznaczający podanie masy zasilania,
  - r0 – komunikat oznaczający wycofanie masy zasilania,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- B drugie napięcie, np. zasilanie:
    - m1 – komunikat oznaczający podanie zasilania,
    - m0 – komunikat oznaczający wycofanie zasilania.

Sygnał taki może przynieść dowolny z agentów A ($A[1 \ldots n]$). Podanie obu napięć powoduje podanie na wyjście do aparatu cewki S sygnału zadziałania kotwicy on/of przez agenta, który jako ostatni przyniósł komunikat zasilania albo masy:

- on – to komunikat do przełączenia kotwicy do stanu przeciwnego do spoczynkowego,
- of[9] – to komunikat do przełączenia kotwicy do stanu spoczynkowego.

Ponieważ dla tak sformułowanego modelu tylko jeden aparat B może podawać sygnał na wejście sygnału zasilania i tylko jeden na wejście sygnału masy C, jeżeli aparatów mogących podawać sygnały zasilania i masy byłoby więcej, to należy zastosować połączenia galwaniczne zbierające sygnały (na rysunku 12 po prawej w ramce).

Uzwojenie steruje aparatem S, przekazując na wyjście sygnał on albo of zależnie od komunikatów r1 albo r0 na wejściu z aparatu C i m1 albo m0 na wejściu z aparatu B. Uzwojenie działa wobec tego jak bramka logiczna AND [72]: wymaga komunikatów m1 i r1 na wejściu aby podać komunikat on na wyjście, w przeciwnym wypadku na wyjściu jest komunikat of. Zakłada się, że w stanie początkowym na wyjściu jest komunikat of, oraz że wycofano oba napięcia (m0 i r0).

Graf automatu modelu uzwojenia przedstawiono na rysunku 13.

Na rysunku 13 zastosowano następujące kolory

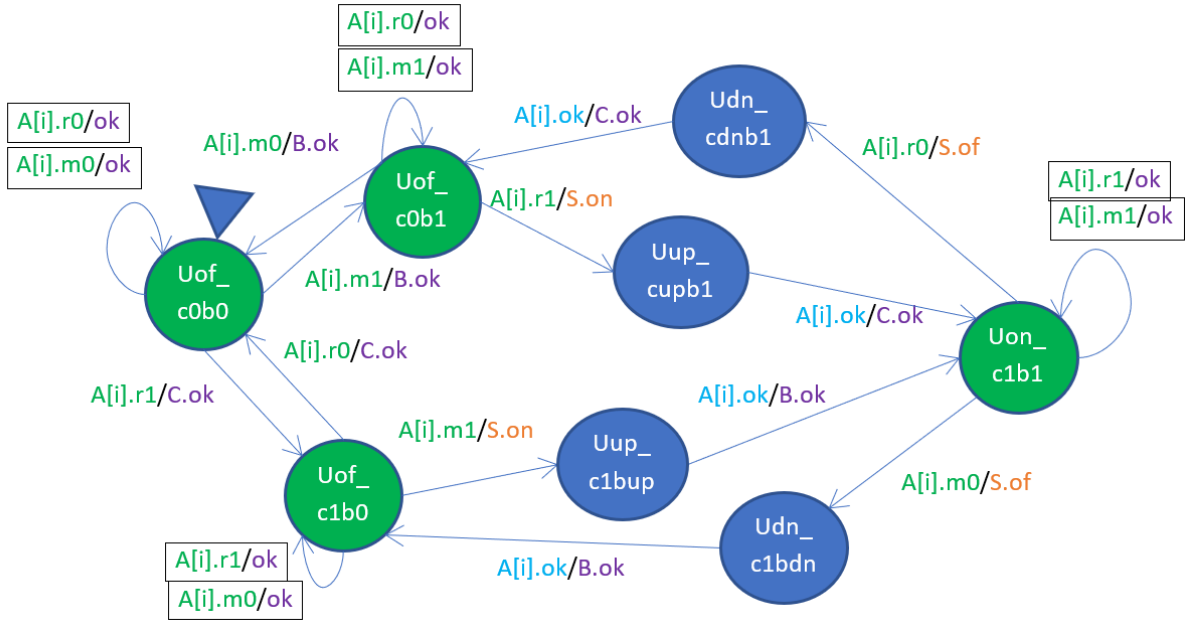
- w opisie akcji:
    - zielony – sygnały wejściowe
    - pomarańczowy – sygnały wyjściowe
    - niebieski – potwierdzenia sygnałów wyjściowych
    - fioletowy – potwierdzenia sygnałów wejściowych. Na sygnał r0/r1 – potwierdzenie kierowane jest (domyślnie) do C, na sygnał m0/m1 – potwierdzenie kierowane jest (domyślnie) do B,
- w oznaczeniu stanów:
    - niebieski – stany niestabilne,

[9] Zastosowano umowne, krótsze oznaczenie of zamiast off.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- zielony – stany stabilne.

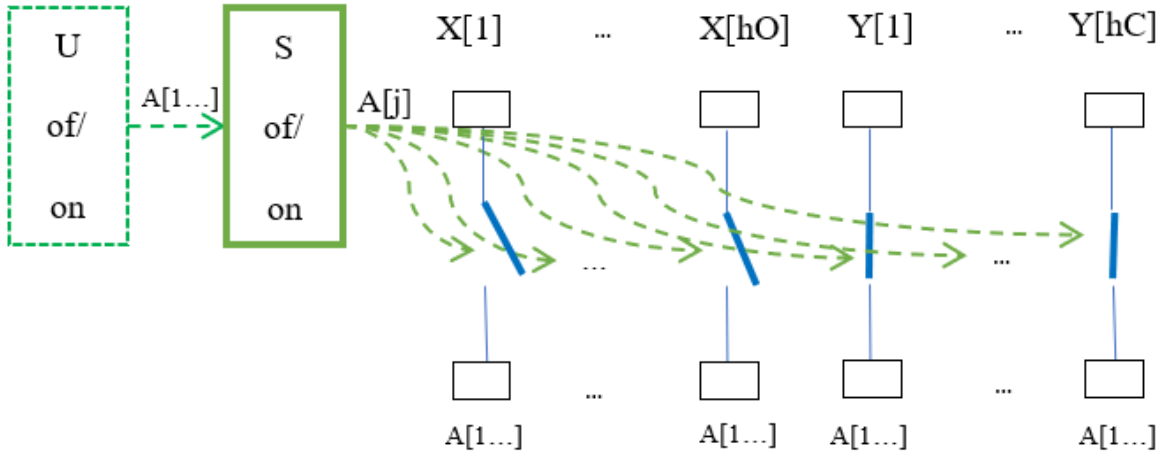
Akcje w czarnych ramkach to sygnały przyjmowane w stanach stabilnych, które nie zmieniają stanu.



Rysunek 13. Graf automatu modelu uzwojenia przekaźnika. Źródło: opracowanie własne na podstawie [35]

4.3 Model kotwicy

Na rysunku 14 przedstawiono schemat poglądowy modelu kotwicy przekaźnika.



Rysunek 14. Schemat poglądowy kotwicy przekaźnika: j – numer agenta kotwicy, hO – liczba zestyków NO, hC – liczba zestyków NC. Źródło: opracowanie własne na podstawie [35]

Kotwica S przekaźnika działa w ten sposób, że przy wzbudzeniu uzwojenia U i podaniu przezeń sygnału on najpierw rozwiera wszystkie zestyki Y (rozwierne, NC), których jest hC, a dopiero

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

potem zwiera wszystkie zestyki X (zwierne, NO), których jest hO. W przeciwnym kierunku, przy odwzbudzeniu uzwojenia U i podaniu przezeń sygnału of najpierw rozwiera wszystkie styki NO, a potem zwiera styki NC. Wejście do stanu rozpoczęcia ruchu kotwicy Sup (przy wzbudzaniu przekaźnika) lub Sdn (przy odwzbudzaniu przekaźnika) uruchamia własnego agenta kotwicy $A[j]$, który wychodzi ze stanu bezczynności (idl) i jest kierowany do kolejnych zestyków tego samego rodzaju (przy przejściu ze stanu Sup do zestyków rozwiernych a potem do zwiernych, przy przejściu ze stanu Sdn do zestyków zwiernych a potem do rozwiernych). Po zakończeniu tych czynności powraca do stanu bezczynności (idl). Zadaniem agenta kotwicy $A[j]$ jest oddziaływanie na każdy określony zestyk w taki sposób, że zmienia on swój stan, a następnie powrót do automatu kotwicy z potwierdzeniem (ok).

Kotwicę można zamodelować inaczej dla:

- przekaźnika zwierającego wyłącznie zestyki zwierne,
- przekaźnika zwierającego wyłącznie zestyki zwierne,

oraz inaczej dla różnej liczby zestyków.

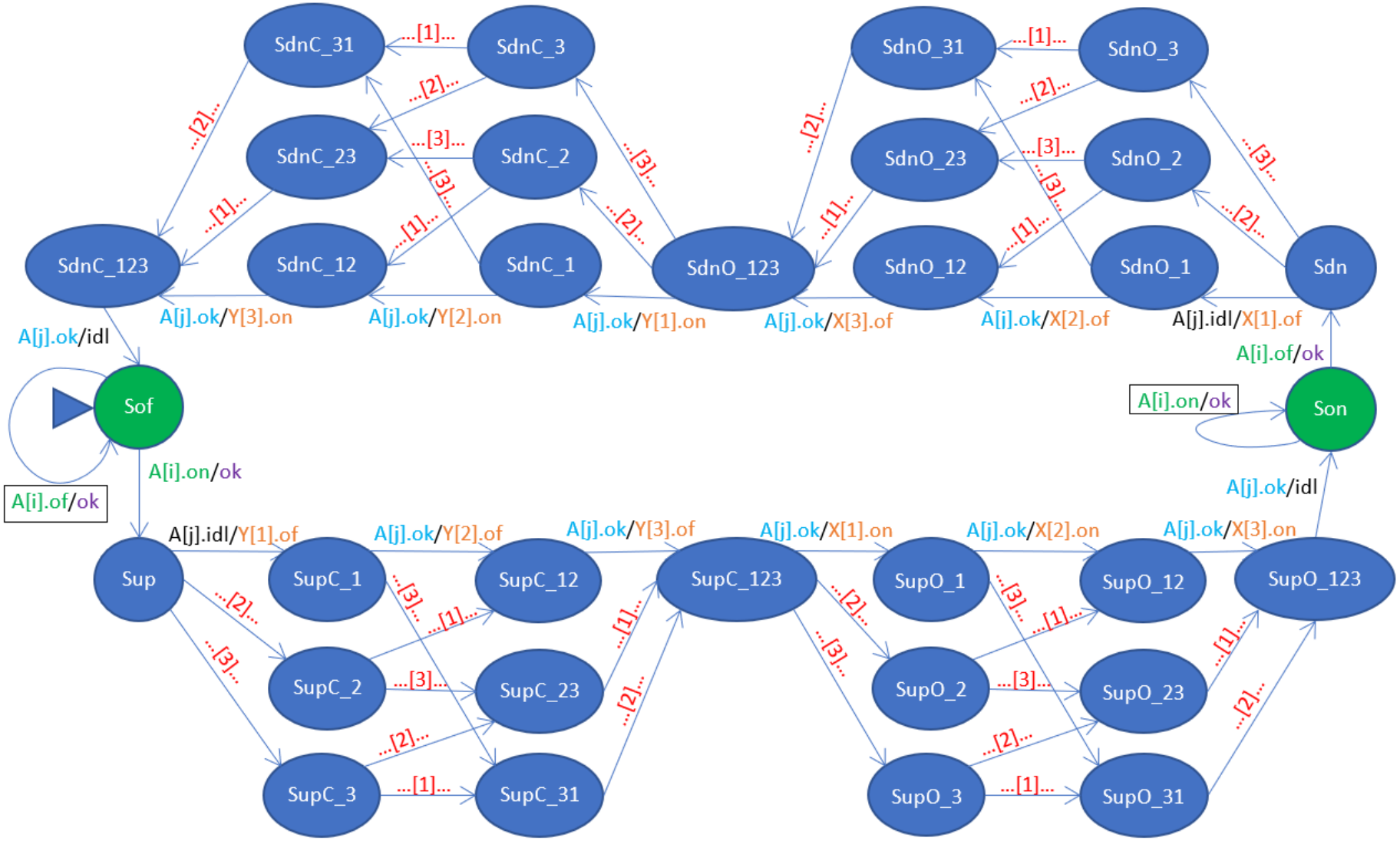
Przedstawiony w rozdziale 5.3 kod prototypu zawiera funkcje pozwalające określić na początku dowolną liczbę zestyków określonego rodzaju. Na rysunku 15 przedstawiono graf modelu kotwicy dla przykładowego przekaźnika zawierającego trzy zestyki zwierne (hO=3) i trzy rozwierne (hC=3).

Na rysunku 15 zastosowano kolory i ramkę takie, jak na rysunku 13. Kolorem czarnym w opisie akcji oznaczono stan bezczynności (idl) agenta kotwicy $A[j]$.

Na rysunku 15 dla poprawy czytelności zastosowano oznaczenia czerwonym kolorem nad łukami opisu będącego powtórzeniem opisu poniżej lub powyżej ze zmianą numeru zestyku wskazaną w nawiasie kwadratowym. Przykładowo łuk pomiędzy stanami Sup a SupC_2 ma opis *...[2]...* . Ponieważ łuk pomiędzy stanem Sup a Sup_C1 ma opis *A[j].idl/Y[1].of*, to łuk pomiędzy stanami Sup a SupC_2 powinien mieć opis *A[j].idl/Y[2].of* (podmianie uległ numer zestyku: z *Y[1]* na *Y[2]*).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

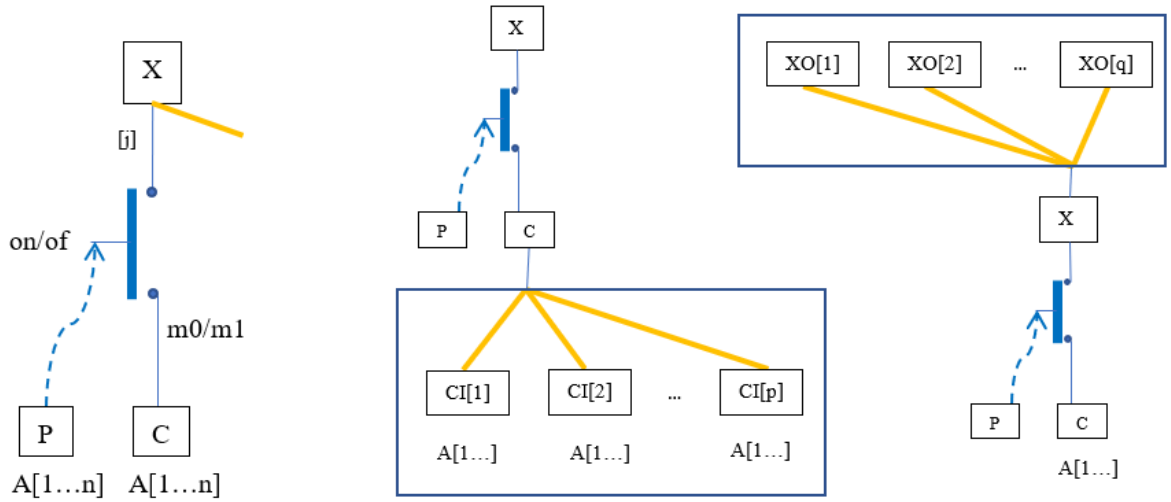
Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology



Rysunek 15. Graf automatu modelu kotwicy przekaźnika. j – numer agenta kotwicy, Y – styki NC, X – styki NO. Źródło: opracowanie własne na podstawie [35]

4.4 Model zestyku

Na rysunku 16 przedstawiono model zestyku.



Rysunek 16. Model zestyku przekaźnika, przełącznika. Źródło: opracowanie własne na podstawie [35]

P jest aparatem powodującym zwarcie sygnałem on albo rozwarcie sygnałem of. W szczególności będzie to:

- kotwica przekaźnika albo
- otoczenie modelu w przypadku zestyku przełącznika sterowanego przez operatora urządzeń (człowieka).

Sygnał on lub of może przynieść dowolny z agentów $A[1 \ldots n]$. C jest aparatem na wejściu. Tylko jeden aparat może być na wejściu tak sformułowanego zestyku. Jeżeli aparatów mogących podawać sygnały na wejście C byłoby więcej, to należy zastosować połączenie galwaniczne zbierające sygnały (po środku rysunku 16 w ramce). Sygnał m z aparatu C to podanie określonego napięcia (m1) lub wycofanie tego napięcia (m0). Typowo jako sygnał m może występować zasilanie (v) lub masa (g), ale mogą występować też inne sygnały.

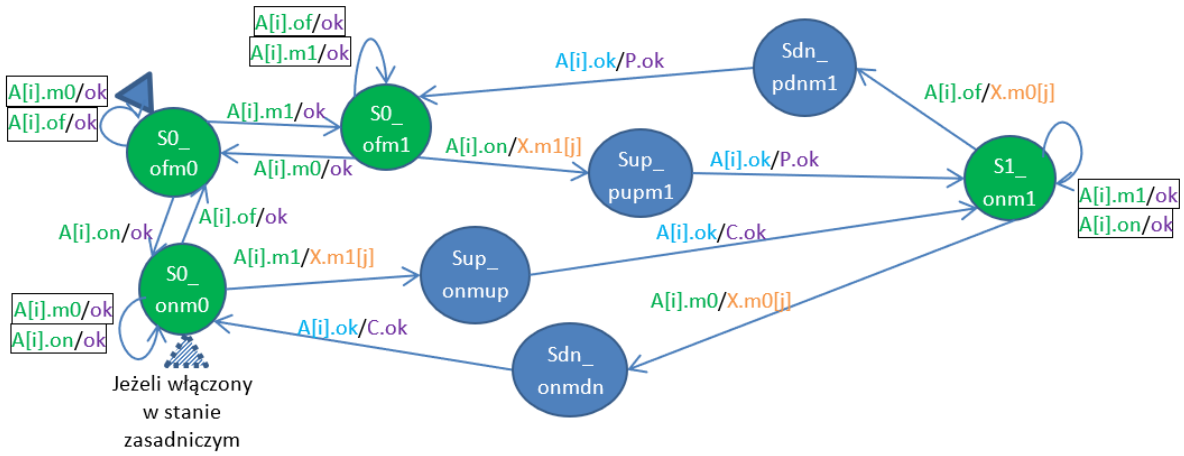
Na wyjściu jest aparat X, którym zestyk steruje, przekazując na wyjście sygnał m1/m0 zależnie od sygnału m1/m0 na wejściu C i stanu styku osiąganego komunikatem on/of. Styk zachowuje się jak bramka logiczna AND, to znaczy wymaga sygnałów on i m1 na wejściu aby podać m1 na wyjście, w przeciwnym wypadku na wyjściu jest m0. Zakłada się, że w stanie początkowym na wyjściu jest m0.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Standardowo (po lewej na rysunku 16) na wyjściu może być tylko jeden aparat - X. Jeżeli zestyk steruje wieloma aparatami, należy zastosować połączenie galwaniczne powielające sygnał (po prawej na rysunku 16 w ramce).

Jeżeli na wyjściu jest połączenie galwaniczne zbierające sygnały, to styk może być j-tym jego wejściem. W przeciwnym wypadku jest jedynym wejściem (kolejnego aparatu) o numerze 1.

Graf automatu modelu zestyku został przedstawiony na rysunku 17.



Rysunek 17. Graf automatu zestyku. Źródło: opracowanie własne na podstawie [35]

Na rysunku 17 zastosowano kolory i ramkę takie, jak na rysunku 13. Dla potwierdzeń sygnałów wejściowych oznaczanych kolorem fioletowym adresat jest domniemany: na sygnał m0/m1 – potwierdzenie do C, na sygnał on/of – potwierdzenie do P.

Są dwie wersje stanu początkowego – oznaczone na rysunku 17 trójkątem dotykającym wierzchołkiem stanu:

- S0_ofm0 dla styku normalnie rozwartego i
- S0_onm0 dla normalnie zwartego.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

4.5 Założenia dla elementów z uszkodzeniami

W rozdziale 7 niniejszej pracy wskazano rodzaje uszkodzeń elementów, z których konstruowane są powiązania z próbą uzasadnienia brania pod uwagę tych rodzajów.

Do każdego modelu sprawnego aparatu dołączony został drugi, uproszczony model aparatu uszkodzonego. Przejście z modelu poprawnego do uszkodzonego następuje poprzez sygnał err agenta uszkodzenia E. Jest to dodatkowy agent, który może przenosić sygnały tak jak pozostałe. Ponieważ dla aparatów na wyjściu jest on nieodróżnialny, gdyż technicznie jest to jeden z agentów wektora A[1..n] o ustalonym numerze, to został oznaczony dla odróżnienia nazwą E.

Nie przewiduje się powrotu do modelu prawidłowo działającego elementu. Gdyby przyjąć odmienne założenie niemożliwe byłoby zaprojektowanie żadnego systemu. Przynajmniej jedna sekwencja działających prawidłowo i nieprawidłowo elementów doprowadzałaby do działania układu tak, jak np. przy przejeździe pociągu, przy faktycznym braku takiej jazdy.

Poniżej przedstawiono dwa przykładowe modele.

4.6 Uszkodzenie uzwojenia przekaźnika

Ten rodzaj uszkodzenia powoduje, że uzwojenie przestaje oddziaływać strumieniem magnetycznym na kotwicę. Oznacza to, że sygnały napięć mogą przychodzić, ale nie powodują ruchu kotwicy do położenia przeciwnego do spoczynkowego. Jeśli przed wystąpieniem uszkodzenia uzwojenie nie było zasilane tak, by przyciągać kotwicę, kotwica pozostaje w tym samym stanie. Jeśli uzwojenie było zasilane i oddziaływało na kotwicę, a ta była w położeniu przeciwnym do spoczynkowego, to przechodzi ona do położenia spoczynkowego przełączając przy tym zestyki. Po wystąpieniu uszkodzenia jedyną funkcją uzwojenia jest potwierdzanie przychodzących sygnałów.

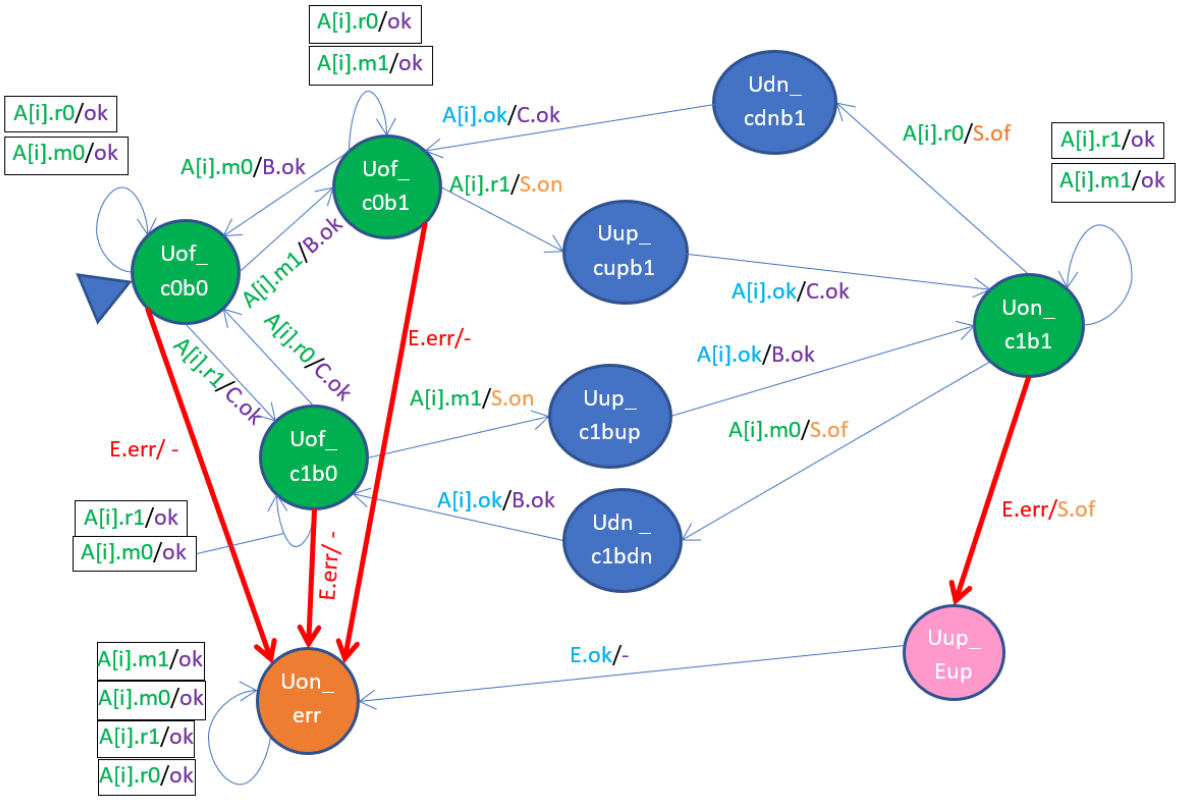
Graf automatu modelu uzwojenia z opisanym uszkodzeniem przedstawiono na rysunku 18.

Na rysunku 18 zastosowano następujące dodatkowe kolory względem tych zastosowanych w grafie sprawnego uzwojenia (przedstawionego na rysunku 13):

- w łukach akcji:
    - czerwony – akcja wywołana przez agenta E, która powoduje wyjście z części grafu odpowiadającej pracy sprawnego urządzenia do części odpowiadającej działaniu po uszkodzeniu,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- w oznaczeniu stanów:
    - pomarańczowy – stany stabilny po wystąpieniu uszkodzenia,
    - różowy – stan niestabilny po wystąpieniu uszkodzenia.



Rysunek 18. Graf automatu modelu uzwojenia z wystąpieniem przerwy. Źródło: opracowanie własne na podstawie [35]

4.7 Spieczenie zestyku zwiernego przekaźnika

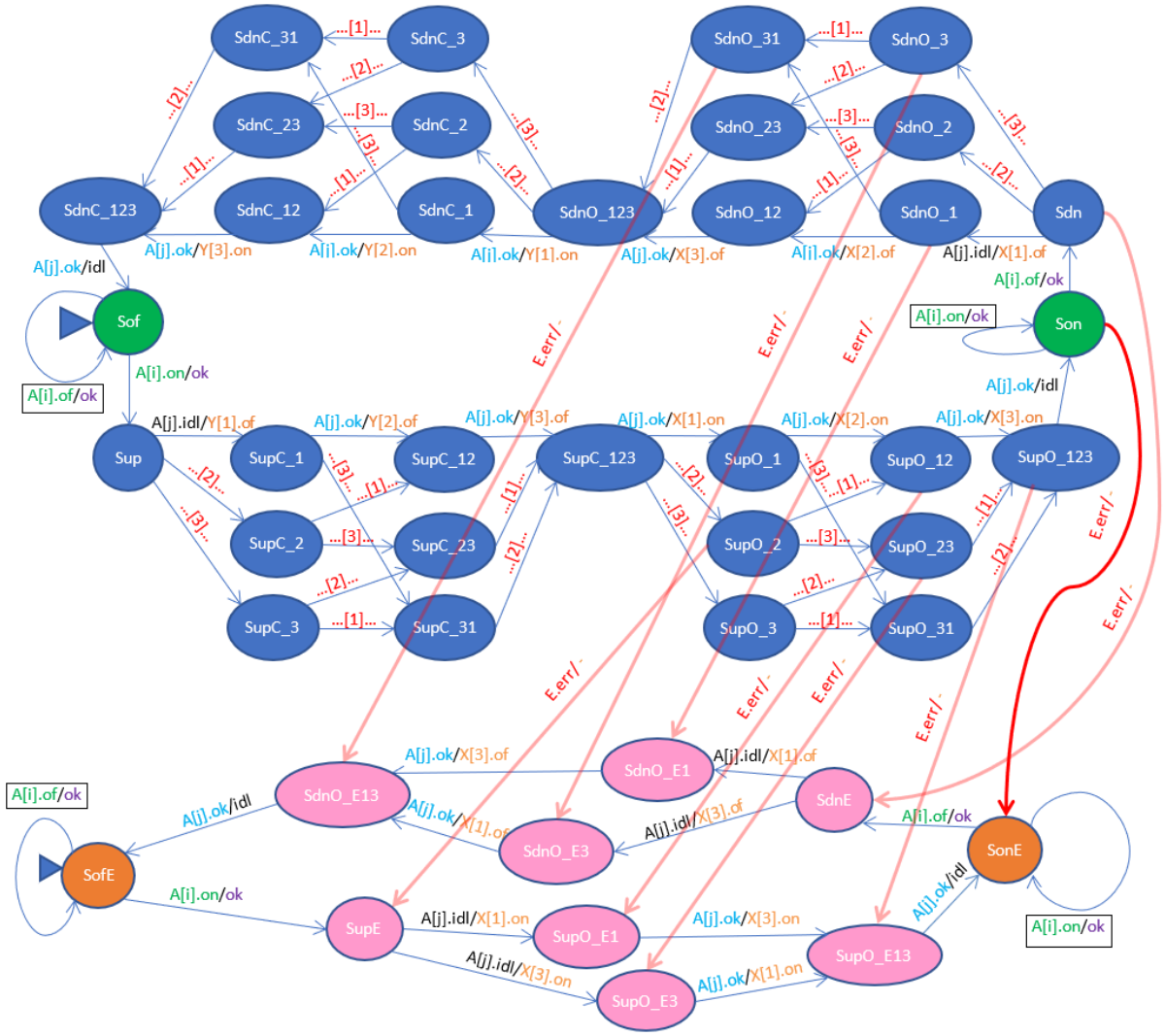
Dla tego typu uszkodzenia wszystkie zestyki zwierne oprócz uszkodzonego działają normalnie, natomiast zestyki rozwierne pozostają stale rozwarte. Uszkodzenie następuje tylko w czasie gdy uszkadzany zestyk zwierny jest zwarty.

Zaproponowany jest model uwzględniający zasadę sztywnego prowadzenia zestyków sformułowaną w fiszce UIC-736R [61] oraz normach IEC-62912 [54][56]. Model ten jest

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

również odpowiedni dla wymagania sztywnego prowadzenia zestyków sformułowanego w normie BN-76 3506-11 [51].

Graf automatu modelu kotwicy z omawianym uszkodzeniem przedstawiono na rysunku 19.



Rysunek 19. Graf automatu modelu kotwicy przekaźnika ze spieczonym zestykiem zwiernym (numer 2). j – numer agenta kotwicy, Y – styki NC, X – styki NO. Źródło: opracowanie własne na podstawie [35]

Na rysunku 19 zastosowano kolory opisane w rozdziale 4.6.

Podstawą rysunku 19 jest rysunek 15 (znajdujący się w rozdziale 4.3) przedstawiający kotwicę przekaźnika z zestykami zwiernymi i rozwiernymi (po trzy) działającą bez uszkodzeń. Rysunek 15 obejmuje stany oznaczone również na rysunku 19 kolorem zielonym i niebieskim. Stany

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

wynikające z działania agenta E zostały oznaczone kolorami pomarańczowymi (stabilne) i różowe (przejściowe).

Do spieczenia zestyku (dowolnego) zwiernego może dojść tylko, gdy przekaźnik jest wzbudzony. Wytypowano jeden z zestyków – ten oznaczony numerem 2. Dlatego łuki różowe i czerwony są kierowane z wybranych stanów odpowiadających sytuacjom, w których zestyk zwierny 2 jest już zwarty. Łuki te kierują do podgrafu znajdującego się poniżej i zawierającego dwa stany stabilne SonE i SofE. W stanie SonE przekaźnik zachowuje się tak, jak przekaźnik sprawny: kotwica utrzymuje swoje położenie, utrzymywana sygnałem on z cewki U, zestyki rozwierne są rozwarte a zwierne zwarte. Przejście z tego stanu do stanu SofE wymaga przejścia przez stany Sdn…, do czego prowadzi przesłanie sygnału of z cewki. Przejście łukami powoduje otwarcie zestyków zwiernych: w dowolnej kolejności i z pominięciem zestyku uszkodzonego, czyli numer 2. Osiągnięty stan SofE jest stanem w którym:

- żaden z zestyków rozwiernych nie został zamknięty,
- zestyk zwierny numer 2 pozostał zamknięty (wskutek nie przemijającego uszkodzenia),
- pozostałe zestyki zwierne (numery 1 i 3) zostały otwarte.

Modelowane działanie odpowiada zasadzie sztywnego prowadzenia zestyków, opisanej na stronie 61: niedopuszczalna jest sytuacja, w której dowolny z zestyków zwiernych przekaźnika i dowolny z rozwiernych pozostaja zamknięte w tym samym momencie, również przy usterce przekaźnika.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

5 Specyfikacja układu przekaźnikowego w języku programowania

5.1 Zasady specyfikacji

5.1.1 Poziomy szczegółowości

Aby móc w programie Dedan wykonać weryfikację układu złożonego z modeli, modele te należy przedstawić w języku programowania IMDS (notacja w widoku serwera [17]). W programie Dedan przeznaczonym do weryfikacji systemów rozproszonych, IMDS jest używany jako język specyfikacji. Język ten wywodzi się z formalnej definicji IMDS. Ten styl specyfikacji jest daleki od typowego stylu programowania. Notacja została przedstawiona szerzej w artykule [17].

Zastosowano typ parametryzowany, pozwalający operować na trzech poziomach szczegółowości: prototypu, typu i instancji. Oznacza to, że np. dla zestyku:

- prototyp – może sterować jednym lub wieloma (n) aparatami, może być po stronie zasilania lub masy,
- typ – jest określone czy znajduje się po stronie masy, czy nie i czy steruje jednym czy więcej aparatem,
- instancja – jest to określony egzemplarz zestyku.

5.1.2 Dołączanie serwerów do modelu

Serwery są dołączane do modelu komendą #INCLUDE. Przykład dołączenia serwera/serwerów do modelu:

```
#INCLUDE C:\...\Arm_NONC.txt parameters(1, 1 ,4, 11, 3)
```

`C:\...\Arm_NONC.txt` – ścieżka pliku zawierającego prototyp.

`parameters(…)` – parametry makrogeneracji[10] serwera/serwerów. Wartości parametrów muszą być zgodne z deklaracjami typów w nagłówku prototypu.

Ten sam prototyp może być dołączany do modelu wielokrotnie, jednak w takim przypadku jego nazwa powinna być parametryzowana żeby uzyskać jednoznaczność nazw w ramach modelu.

[10] Makrogenerator jest narzędziem skojarzonym ze środowiskiem Dedan, umożliwiającym wydajne generowanie kodu programu na podstawie parametrów zadanych w prototypach.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Nagłówek serwera zawiera deklaracje zmiennych.

5.1.3 Nagłówek prototypu i typy danych

Nagłówek prototypu ma postać `parameters(…)`, na przykład:

```
parameters(@h=num,@m=text,@n=num,@t=num).
```

W nagłówku występują parametry (będące jednocześnie zmiennymi globalnymi prototypu) oraz ich typy. Jako typy danych można wyróżnić m.in.:

- `text` – skalar tekstowy, przykład wywołania: `”v”`,
- `num` – skalar numeryczny, przykład wywołania: `5`.

Deklaracje zmiennych poprzedzane są symbolem `@`.

Oprócz typów wymienionych dla nagłówka, mogą występować typy pochodne.

W prototypie może występować wiele serwerów. Parametry prototypu mogą być użyte w nagłówkach serwerów, na przykład:

```
server:    join@t(servers C[@h],F;agents A[@n]),
```

parametr `@t` jest użyty do nadania serwerowi unikalnej nazwy, `@h` specyfikuje rozmiar wektora dostępnych innych serwerów, `@n` specyfikuje rozmiar wektora dostępnych agentów.

5.1.4 Zmienne

Wewnątrz prototypu można powoływać zmienne. Zmiennymi są:

- parametry prototypu,
- zmienne globalne,
- zmienne lokalne,
- iteratory (zmienne wierszowe).

Nazwy zmiennych mogą składać się z liter i cyfr.

Zmienne globalne występują przed frazą `server` (lub `agent`) i są ważne do końca pliku. Do tej kategorii zalicza się również parametry prototypu. W czasie makrogeneracji prototypu muszą mieć ustalone wartości, pochodzące z wywołania (wartości parametrów aktualnych) lub literały i funkcje zwracające wartość. Przykład: `@m=$elseq(@a,@i)` – zmienna `@m` otrzymuje wartość `@i`-tego elementu sekwencji `@a`.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Zmienne lokalne występują w nagłówku serwera (lub agenta) po nazwie typu lub w treści serwera, np.:

```
server: X@g=text() @h=num() @j=seq_n(@h,2) (servers A,B,C; agents P,Q,R) …;
```

i są ważne do średnika kończącego definicję typu.

Zmienne wierszowe występują przed definicją stanu, usługi lub akcji wyłącznie jako iteratory (w nawiasach kątowych). Zmienne wierszowe są ważne do przecinka kończącego definicję stanu, usługi lub akcji.

Przykład a) wskazuje iterację stanów przy pomocy indeksu, b) iterację w nazwie stanu, c) iterację akcji w której występuje stan wejściowy mający w nazwie `@h` zer a stan wyjściowy mający ciąg bitowy o tej samej długości w którym jeden bit zamieniono na 1, d) iterację akcji w których na wejściu i wyjściu występuje element sekwencji (lub zbioru) `@s`.

a) `<@a=$subset(1..@h,1,1)>@m0[@a]`,
b) `<@a=$subset(1..@h,1,1)>@m@a`,
c) `<@a=$bits(@h,0)><@b=$nbits(@h,0,1,1)> {message, state@a} -> {message, state@b}`,
d) `<@a=@s> {message1, state@a} -> {message2, state@a}`.

Składnia akcji jest przedzielona strzałką (`->`) zastępującą literę λ. Określenie `this` odnosi się do tego serwera dla którego jest wykonywany opis akcji.

Iterator przebiega wszystkie elementy swojego zakresu. Jeżeli występuje wiele niezależnych iteratorów, to tworzą iloczyn kartezjański swoich wartości. Częściej kolejne iteratory są generowane ze zbioru wartości poprzednich, wtedy tworzą porządek leksykograficzny, na przykład:

- iterator `<@a=$elseq(@x)>` dla sekwencji `@x=(1..3)`, da wartości `1, 2, 3`
- iterator `<@a=$elseq(@x,2,1)>` dla sekwencji `@x=( (1,2),(4,3) )` da wartość `4`
- iterator `<@a=$subset(@x,1,2)>` dla sekwencji `@x=(1..3)` da wartości `(1) (2) (3) (1,2) (2,3) (1,3)`
- iterator `<@a=$nbits(3,0,2,2)>` da wartości `110 101 011` (generuje ciąg `3` bitów `0`, wśród których od `2` do `2` jest zastępowanych przez `1`).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

5.1.5 Wartości literalne

W nadawaniu wartości zmiennym i w parametrach funkcji można użyć wartości literalnych na przykład:

- Wartość numeryczna całkowita, np. `2`
- Wartość tekstowa, np. `abc`
- Sekwencja, np. `(1,2,3)` `(1..3),` `(1..2,3)`. Podobnie dla podzbioru, tylko elementy nie mogą się powtarzać.
- Ciąg bitowy `0001`

5.1.6 Odwołania do zmiennych

Odwołanie do zmiennej jest dokonywane poprzez podanie jej nazwy: `@a`. Jeżeli odwołanie jest w tekście nieiterowanym, to dla skalara jest wstawiana jego wartość, a dla typów złożonych (sekwencje, zbiory) skonkatenowane wszystkie elementy. W obrębie działania iteratora jest brana jego aktualna wartość, na przykład wiersz `<@a=(1,2)> @a` wygeneruje dwa wiersze:

- `1`
- `2`

5.1.7 Funkcje

Funkcje generują wartości i zestawy wartości. Wyróżnić można funkcje działające na liczbach, na zbiorach (wybierają element lub elementy ze zbiorów), na sekwencjach (generują określone sekwencję), na ciągach bitowych (np. zmieniają jedną 1 na 0 w ciągu). Przykładowe funkcje preprocesora:

- `$subseq(a,i,j)` - `a` – parametr będący sekwencją, `i,j` – parametry numeryczne. Zwraca podsekwencję; można użyć tylko w iteratorze
    - przykład odwołania: `<@a=$elseq(@b)>` dla `@b=seq(1..3)` , da wiersze zawierające `1 2 3`
- `$nbits(i,0/1,j, k)` - `i,j,k` – parametry numeryczne, generuje każdy ciąg i bitów 0/1 wśród których jest od j do k bitów 1/0
    - przykład odwołania: `<@a=$nbits(3,0,1,2)>` da wiersze zawierające `100 010 001 110 101 011.`

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

5.1.8 Inne podstawowe elementy języka opisu prototypu

W treści prototypu w pierwszej kolejności definiowane są typy serwerów (po słowie `server`: w taki sposób, jak zostało to zaprezentowane w podrozdziałach dotyczących nagłówków prototypu.

Następnie, po słowie `states` są wymienione w postaci zbioru (w nawiasie klamrowym, po przecinku) nazwy stanów prototypu. Nazwy te można odnaleźć w wierzchołkach grafów modeli elementów składowych przedstawionych na rysunkach w rozdziale 4.

Następnie, po słowie `services` są wymienione w postaci zbioru (w nawiasie klamrowym, po przecinku) usługi serwera.

Następnie, po słowie `actions` są wymienione w postaci zbioru (w nawiasie klamrowym, po przecinku) działania (akcje) serwera.

W opisie akcji są wykorzystywane symbole `->` (myślnik i znak większości) oznaczający akcję (w łukach w rozdziale 2.8 i w grafach na rysunkach w rozdziale 4 użyto w tym celu znaku ukośnika `/`). Po obu stronach symboli `->` w nawiasie klamrowym jest zawarta para rozdzielona przecinkiem:

- rozdzielone znakami kropki: agent, serwer (`this` oznacza serwer, którego dotyczy opis prototypu), usługa,
- rozdzielone znakami kropki: serwer i stan.

W kolejnym rozdziale zostaną przedstawione przykłady wywołania, nagłówków i treści specyfikacji prototypów modeli, które w rozdziale 4 były opisane grafami.

5.2 Specyfikacja uzwojenia przekaźnika

5.2.1 Przykład wywołania

```
#INCLUDE C:\...\Coil.txt parameters("v","g",2,1)
```

Gdzie parametry wywołania są następujące:

`„v”` – napięcie wyższe z wejścia B (zazwyczaj Vcc – `„v”`),

`„g”` – napięcie niższe z wejścia C (zazwyczaj Gnd – `„g”`),

`2` – liczba agentów sygnałowych w systemie: otoczenie (przyciski), kotwice przekaźników i agenty uszkodzeń,

`1` – numer egzemplarza uzwojenia.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

5.2.2 Nagłówek prototypu

```
parameters(@m=text,@r=text,@n=num,@t=num)
```

Gdzie poszczególne zmienne oznaczają:

`@m` – id sygnału z jednej strony (v,g) - parametr B serwera; `@m=”v”` w wywołaniu powyżej,

`@r` – id sygnału z drugiej strony (v,g) - parametr C serwera; `@r="g"` w wywołaniu powyżej,

`@n` – liczba agentów sygnałowych w systemie; `@n=2` w wywołaniu powyżej,

`@t` – numer egzemplarza typu; `@t=1` w wywołaniu powyżej.

5.2.3 Nagłówek serwera

```
server:    coil@t(servers B,C,S;agents A[@n])
```

Gdzie:

`coil@t` – nazwa serwera,

`@t` – odwołanie do parametru prototypu (numer egzemplarza),

`B` – serwer sterujący podaniem napięcia wyższego (zazwyczaj Vcc),

`C` – serwer sterujący podaniem napięcia niższego (zazwyczaj Gnd),

`S` – serwer aparatu kotwicy,

`A[@n]` – wektor agentów przenoszących sygnały,

`@n` – odwołanie do parametru prototypu (liczba agentów sygnałowych w systemie).

5.2.4 Treść prototypu

```
server:    coil@t(servers B,C,S;agents A[@n]),
states {Uof_c0b0, Uof_c0b1, Uof_c1b0, Uon_c1b1, Uup_c1bup,
Uup_cupb1, Udn_c1bdn, Udn_cdnb1},
services {@m_0[1], @m_1[1], @r_0[1], @r_1[1], ok},
actions {
```

Sekcja `actions` obejmuje łuki określone w grafie danego elementu. W tym przypadku jest to cewka przekaźnika, której graf znajduje się na rysunku 13.

Poniższa sekcja dotyczy stanu stabilnego opisanego jako wierzchołek Uof_c0b0 grafu. Jest to stan, w którym cewka nie wysyła sygnału do kotwicy (Uof) dlatego, że nie otrzymuje napięcia z wejść (c0 i b0).

```
//stable state Uof_c0b0
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
<i=1..@n>{A[i].this.@m_0[1], this.Uof_c0b0} -> {A[i].B.ok,
this.Uof_c0b0},
<i=1..@n>{A[i].this.@r_0[1], this.Uof_c0b0} -> {A[i].C.ok,
this.Uof_c0b0},
<i=1..@n>{A[i].this.@m_1[1], this.Uof_c0b0} -> {A[i].B.ok,
this.Uof_c0b1},
<i=1..@n>{A[i].this.@r_1[1], this.Uof_c0b0} -> {A[i].C.ok,
this.Uof_c1b0},
```

Pierwsze dwa wiersze obejmują łuk grafu opatrzony opisami w ramkach: dowolny z agentów A[i] (jest ich od 1 do @n) przynosi komunikat m0 (pierwszy wiersz) lub r0 (drugi), co powoduje łuk będący pętlą – pozostanie w tym samym stanie. Za każdym razem agent, który przybył z komunikatem z określonego źródła, po wywołaniu łuku wraca do tego źródła z komunikatem ok (m0, m1 do B, r0, r1 do C). Jest to zgodne z opisami przyjętymi w rozdziale 4.2 i na rysunku 12.

Kolejne dwa określają dwa łuki do stanów Uof_c0b1 (po przyniesieniu przez agenta komunikatu m1) i Uof_c1b0 (po przyniesieniu przez agenta komunikatu r1).

```
// stable state Uof_c0b1
<i=1..@n>{A[i].this.@m_0[1], this.Uof_c0b1} -> {A[i].B.ok,
this.Uof_c0b0},
<i=1..@n>{A[i].this.@r_0[1], this.Uof_c0b1} -> {A[i].C.ok,
this.Uof_c0b1}
<i=1..@n>{A[i].this.@m_1[1], this.Uof_c0b1} -> {A[i].B.ok,
this.Uof_c0b1},
<i=1..@n>{A[i].this.@r_1[1], this.Uof_c0b1} -> {A[i].S.on,
this.Uup_cupb1},
// stable state Uof_c1b0
<i=1..@n>{A[i].this.@m_0[1], this.Uof_c1b0} -> {A[i].B.ok,
this.Uof_c1b0}
<i=1..@n>{A[i].this.@r_0[1], this.Uof_c1b0} -> {A[i].C.ok,
this.Uof_c0b0},
<i=1..@n>{A[i].this.@m_1[1], this.Uof_c1b0} -> {A[i].S.on,
this.Uup_c1bup},
<i=1..@n>{A[i].this.@r_1[1], this.Uof_c1b0} -> {A[i].C.ok,
this.Uof_c1b0},
```

Powyższe dwa stany, są również stabilne. Zawierają także pętle i możliwość powrotu do stanu Uof_c0b0.

W dalszej części zostają określone łuki prowadzące ze stanów stabilnych omówionych wcześniej poprzez stany niestabilne (Uup) do stanu stabilnego Uon_c1b1. Łuki do stanu

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

niestabilnego zawierają komunikaty on kierowane do kotwicy S. Łuki do stanu stabilnego Uon_c1b1 zawierają komunikaty ok agenta wysłanego do kotwicy (pierwszy komunikat) i do źródła, które jako ostatnie wywołało wyjście ze stanu stabilnego Uof (B dla Uup_c1bup, C dla Uup_cupb1).

```
//path from of to on – b
<i=1..@n>{A[i].this.ok, this.Uup_c1bup} -> {A[i].B.ok,
this.Uon_c1b1},
//path from of to on – c
<i=1..@n>{A[i].this.ok, this.Uup_cupb1} -> {A[i].C.ok,
this.Uon_c1b1},
```

Pozostanie lub wyjście ze stanu stabilnego Uon_c1b1 zostało opisane na podobnych zasadach, jak dla stanów Uof. Stan Uon_c1b1 to stan, w którym kotwica została uniesiona.

```
//stable state Uon_c1b1
<i=1..@n>{A[i].this.@m_0[1], this.Uon_c1b1} -> {A[i].S.of,
this.Udn_c1bdn},
<i=1..@n>{A[i].this.@r_0[1], this.Uon_c1b1} -> { A[i].S.of,
this.Udn_cdnb1},
<i=1..@n>{A[i].this.@m_1[1], this.Uon_c1b1} -> {A[i].B.ok,
this.Uon_c1b1},
<i=1..@n>{A[i].this.@r_1[1], this.Uon_c1b1} -> {A[i].C.ok,
this.Uon_c1b1},
```

Łuki prowadzące ze stanu stabilnego Uon_c1b1 do stanów Uof_c0b1 i Uof_c1b0 zostały opisane na podobnych zasadach, jak dla łuków w przeciwnym kierunku opisanych wcześniej.

```
//path from on to of – b
<i=1..@n>{A[i].this.ok, this.Udn_c1bdn} -> {A[i].B.ok,
this.Uof_c1b0},
//path from on to of – c
<i=1..@n>{A[i].this.ok, this.Udn_cdnb1} -> {A[i].C.ok,
this.Uof_c0b1},
};
```

Na podobnych zasadach sformułowano treści pozostałych prototypów, dla których w dalszych punktach wskazano jedynie ich nagłówki i sposób wywołania.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

5.3 Specyfikacja kotwicy przekaźnika

5.3.1 Przykład wywołania

```
#INCLUDE C:\...\ArmNONC.txt parameters(4,5,2,3,1)
```

Gdzie parametry wywołania są następujące:

4 – liczba styków NO,

5 – liczba styków NC,

2 – numer agenta kotwicy,

3 – liczba agentów (razem: agent otoczenia, agenci kotwic przekaźników i agenci uszkodzeń),

1 – numer egzemplarza serwera w ramach typu armNONC.

5.3.2 Nagłówek prototypu

```
parameters(@hO=num,@hC=num,@j=num,@n=num,@t=num)
```

Gdzie poszczególne zmienne oznaczają:

@hO – liczba styków NO, np. 4,

@hC – liczba styków NC, np. 5,

@j – numer agenta kotwicy, np. 2,

@n – liczba agentów sygnałowych w systemie, np. 3,

@t – numer egzemplarza serwera w ramach typu armNO, np. 1.

5.3.3 Nagłówek serwera

```
server:    armNONC@t(servers U,X[@hO],Y[@hC];agents A[@n]),
```

Gdzie:

armNONC@t – nazwa serwera, @t – odwołanie do parametru prototypu (numer egzemplarza),

U – serwer uzwojenia sterujący przełączeniem kotwicy,

X[@hO] – serwery styków zwiernych NO sterowanych przez kotwicę, @hO – liczba styków NO, odwołanie do parametru prototypu,

Y[@hC] – serwery styków rozwiernych NC sterowanych przez kotwicę, @hC – liczba styków NC, odwołanie do parametru prototypu,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

A[@n] – wektor agentów przenoszących sygnały, @n – odwołanie do parametru prototypu (liczba agentów sygnałowych w systemie).

5.4 Specyfikacja zestyku

5.4.1 Przykład wywołania

```
#INCLUDE C:\..\Contact.txt parameters(2,”g”,2,1)
```

Gdzie parametry wywołania są następujące:

2 – aparat jest drugim źródłem sygnału dla aparatu wyjściowego,

„g” – nazwa napięcia (w tym przypadku masa),

2 – liczba agentów sygnałowych w systemie: otoczenie (przyciski), kotwice przekaźników i agenty błędów,

1 – numer egzemplarza zestyku.

5.4.2 Nagłówek prototypu

```
parameters(@j=num,@m=text,@n=num,@t=num)
```

Gdzie poszczególne zmienne oznaczają:

@j – którym numerem źródła jest styk dla aparatu wyjściowego, np. 2,

@m – id sygnału („v”, „g” lub inny), np. "v",

@n – liczba agentów sygnałowych w systemie, np. 3,

@t – numer egzemplarza serwera w ramach prototypu contact, np.1.

5.4.3 Nagłówek serwera

```
server:    contact@t(servers P,C,X;agents A[@n])
```

Gdzie:

contact@t – nazwa serwera,

@t – odwołanie do parametru prototypu (numer egzemplarza),

P – serwer sterujący przełączeniem styku,

C – serwer podający napięcie,

X – serwer aparatu sterowanego przez styk,

A[@n] – wektor agentów przenoszących sygnały,

@n – odwołanie do parametru prototypu (liczba agentów sygnałowych w systemie).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

6 Przebieg badania powiązania

6.1 Założenia

Podstawowym badaniem jest sprawdzenie, czy przy wszystkich elementach działających poprawnie, i przy sekwencji zwarć/rozwarć styków sterowanych z zewnątrz, układ osiąga wyznaczony stan końcowy. W przypadku badanego powiązania reakcją jest wzbudzenie określonych przekaźników. Prawidłowość działania w tym zakresie wynika z zasad prowadzenia ruchu i obsługi urządzeń srk przez personel. Zasadniczo, do zwalidowania modelu można brać pod uwagę pracę urządzeń podczas normalnego ruchu i normalnej obsługi. Dalsze eksperymenty powinny natomiast wykazać prawidłowość rozwiązania podczas:

- Nieprawidłowej obsługi,
- Nieprawidłowej jazdy (np. wstecz),
- Możliwych usterek występujących w elementach układu (np. przerwa zestyku, cewki).

Kolejnym badaniem powinno być ponownie sprawdzenie reakcji (osiągnięcia zamierzonego stanu końcowego) przy więcej niż jednej usterce, jeżeli pojedyncza nie ujawnia się w cyklu pracy.

W typowej praktyce inżynierskiej, poprawne działanie systemu jest wykazywane poprzez manualną analizę, przypominającą wodzenie palcem po układzie i wskazywanie sekwencji wzbudzeń i zwarć. Nie można jednak tego uznać za dowód poprawności, gdyż nie sposób w manualny sposób wykazać nieuchronności właściwej terminacji. Weryfikacja modelowa sprawdza wszystkie możliwe ścieżki zachowania układu jednocześnie, co prowadzi do dowodu poprawności. Należy przy tym rozróżnić, że formalny dowód sporządzony przez weryfikator dotyczy wszystkich zachowań, zaś świadek (pojęcie wyjaśnione w rozdziale 2.4) otrzymany jako wynik przedstawia jedną, przykładową ścieżkę zachowania układu. Inne ścieżki można otrzymać poprzez symulację układu, dostępną w środowisku weryfikacji.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

6.2 Przykład powiązania

Przykład dotyczy fragmentu powiązania blokady liniowej Eap z urządzeniami stacyjnymi. Całe powiązanie przedstawiono na rysunku 20. Blokada i urządzenia stacyjne pracują względem siebie asynchronicznie. Można określić wejścia i wyjścia (są równoległe dla wielu sygnałów) na podstawie dostępnej specyfikacji interfejsu systemu komputerowego i dokumentów opisujących blokadę (rozwiązanie przekaźnikowe). Interesujący fragment powiązania, który będzie podlegać dalszemu modelowaniu dotyczy obwodu oddziaływania pociągu, czyli obwodu blokady, który:

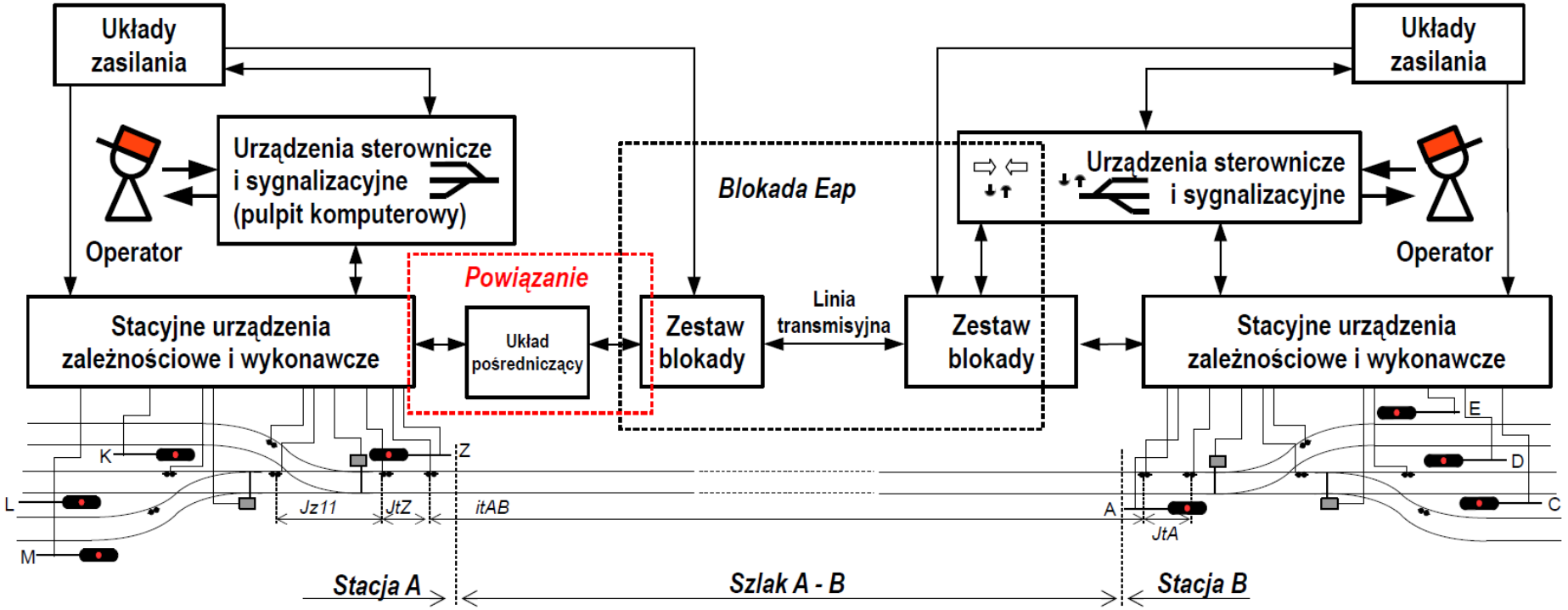
- otrzymuje informacje:
    - z urządzeń stacyjnych:
        - o zajętości i niezajętości wybranych odcinków kontroli (it, Jt, Jz), na które został podzielony układ torowy stacji i szlaku,
        - w postaci polecenia DKO obsługiwanego z pulpitu urządzeń stacyjnych,
        - o wskazaniu semafora wjazdowego (Kc, Kpz),
    - z blokady o ustawionym kierunku na wjazd (Dp),
- przekazuje informacje do urządzeń stacyjnych i blokady o stanie przekaźników W, Z i dKo.

Blokada Eap istnieje w wielu wersjach (m.in. [38][43]). Przyjęta w rozdziale 6 wersja jest oparta na rozwiązaniach zbieżnych z albumowymi [38].

Zastosowano przypadek bez posterunku SKP (stwierdzenia końca pociągu) i z jedną nastawnią (dysponującą) na stacji A.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

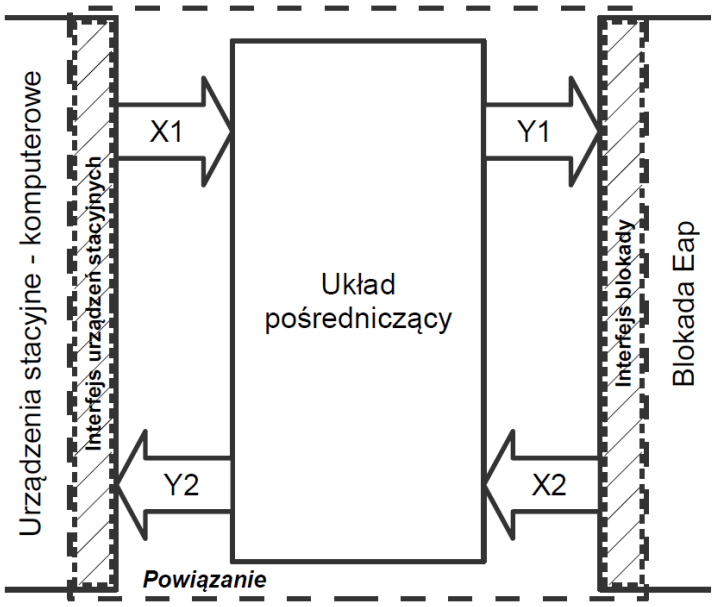
Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology



Rysunek 20. Schematyczny układ torów i związana z nim lokalizacja urządzeń i powiązania. K, L, M, Z – semafory stacji A istotne dla powiązania, Jz11, JtZ, itAB, JtA – odcinki kontroli niezajętości istotne dla powiązania. Źródło: opracowanie własne na podstawie [35]

6.3 Opis układu pośredniczącego i wektorów sygnałów

Na rysunku 21 przedstawiono ogólny szkic współpracujących systemów: komputerowego i blokady, bazujący na rysunku przedstawionym w rozdziale 1.2.



Rysunek 21. Ogólny szkic współpracujących systemów: komputerowego i blokady Eap. Źródło: opracowanie własne na podstawie [40]

Wektor **X1** jest wektorem sygnałów wyjściowych z urządzeń stacyjnych.

Wektor **X2** jest wektorem sygnałów wyjściowych z blokady.

Wektor **Y1** jest wektorem sygnałów wejściowych do blokady.

Wektor **Y2** jest wektorem sygnałów wejściowych do urządzeń stacyjnych.

Wektory **X1** i **X2** są jednocześnie wektorami wejściowymi układu pośredniczącego, wektory **Y1** i **Y2** jego wektorami wyjściowymi.

Wszystkie zmienne wektorów **X1**, **X2**, **Y1** i **Y2** są binarne. 0 oznacza brak zwarcia obwodu albo wciśnięcia przycisku i wskutek tego zwarcia obwodu. 1 oznacza zwarcie obwodu albo wciśnięcie przycisku i wskutek tego zwarcie obwodu. 0 oznacza brak napięcia o wymaganej wartości na danym wejściu lub wyjściu, 1 jego obecność.

Wszystkie zmienne są zmiennymi w czasie.

Sygnały są następujące (znak / poprzedzający zmienną oznacza negację):

1. Blokada Eap (na podstawie schematów blokady, które również można odnaleźć w albumie [43]):

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

a. Wektor **X2** sygnałów wyjściowych z blokady (i jednocześnie wejściowych do układu pośredniczącego od strony blokady):
    i. Sygnały wykorzystywane w obwodach zależnościowych:
        1. Połączenie szeregowe zestyków przekaźników zwiernych Op, Dp, Pwl i rozwiernego Wbl (w obwodzie przekaźnika sygnałowego S) – *x2 1*
        2. Połączenie szeregowe zestyków przekaźników zwiernych Op, Dp, pS i rozwiernego Wbl (w obwodzie przekaźnika sygnałowego S) – *x2 2*
        3. Zestyk rozwierny przekaźnika Pwl (w obwodzie świateł semafora wyjazdowego na szlak). – *x2 3*
        4. Zestyk rozwierny przekaźnika W (w obwodzie przekaźnika sygnałowego S semafora wjazdowego ze szlaku) – *x2 4*
    ii. Sygnały przekazywane przez pulpit (układ zestyków zasilających następujące żarówki i lampki):
        1. Żarówka biała strzałki przyjazd (ciągłe lub migające) – *x2 5*
        2. Żarówka czerwona strzałki przyjazd (ciągłe lub migające) – *x2 6*
        3. Żarówka biała strzałki odjazd (ciągłe lub migające) – *x2 7*
        4. Żarówka czerwona strzałki odjazd (ciągłe) – *x2 8*
        5. Lampka biała Ko (ciągłe lub migające) – *x2 9*
        6. Lampka czerwona Pwl (ciągłe) – *x2 10*
        7. Lampka biała dKo (ciągłe) – *x2 11*
        8. Licznik dKo (ciągłe) – *x2 12*
        9. Licznik dPo (ciągłe) – *x2 13*
        10. Dzwonek (ciągłe) – *x2 14*
b. Wektor **Y1** sygnałów wejściowych do blokady (i jednocześnie wyjściowych z układu pośredniczącego do blokady):
    i. Sygnały z zestyków przekaźników:
        1. Zestyk rozwierny przekaźnika Pml sygnałowego pomocniczego liniowego semaforów wyjazdowych– *y1 1*
        2. Zestyk zwierny przekaźnika Pml sygnałowego pomocniczego liniowego semaforów wyjazdowych – *y1 2*
        3. Połączenie szeregowe zestyków rozwiernych przekaźników sygnałowych S, SSz i zestyków zwiernych przekaźników it

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

kontroli niezajętości toru szlakowego i Kc semafora wjazdowego – *y1 3*

4. Połączenie szeregowe zestyków rozwiernych przekaźników it i Jt oraz zwiernego przekaźnika Jz – *y1 4*
5. Połączenie szeregowe zestyków zwiernych przekaźników it i Jt oraz rozwiernego przekaźnika Jz– *y1 5*
6. Połączenie szeregowe zestyków rozwiernego przekaźnika Kc i zwiernego Kpz – *y1 6*
7. Zestyk zwierny przekaźnika it– *y1 7*

ii. Sygnały z pulpitu

1. Wbl wciskany – *y1 8*
2. Wbl odciągany (oWbl) – *y1 9*
3. Poz – *y1 10*
4. Ko – *y1 11*
5. dKo – *y1 12*
6. dPo – *y1 13*

2. urządzenia stacyjne (na podstawie specyfikacji interfejsu opisanej w rozdziale 6.5):
   a. Wektor **X1** sygnałów wyjściowych z urządzeń stacyjnych (i jednocześnie wejściowych do układu pośredniczącego od strony tych urządzeń):
      i. Sygnały zależnościowe:
         1. S1_Kc – *x1 1*
         2. S2_Kpz – *x1 2*
         3. S3_/Sz, nie wyświetlony Sz na semaforze wjazdowym – *x1 3*
         4. S4_Jt, pierwszy odcinek kontroli niezajętości za semaforem wjazdowym wolny – *x1 4*
         5. S5_Jz, drugi odcinek kontroli niezajętości za semaforem wjazdowym wolny – *x1 5*
         6. S6_/Uwy, utwierdzona droga przebiegu wyjazdowego – *x1 6*
         7. S7_it, odcinek pokrywający szlak wolny – *x1 7*
      ii. Polecenia zmiany stanu[11]:
         1. DKO – *x1 8*
         2. DPO – *x1 9*

[11] Wykorzystano nazwy poleceń pisane wielkimi literami, aby odróżnić niektóre z nich od nazw przekaźników blokady.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

3. KO – *x1 10*
4. OWBL – *x1 11*
5. PZK – *x1 12*
6. WBL – *x1 13*

b. Wektor **Y2** sygnałów wejściowych do urządzeń stacyjnych (i jednocześnie wyjściowych z układu pośredniczącego do urządzeń stacyjnych):

i. Sygnały zależnościowe

1. K1_ZNWy, zezwolenie na ustawienie przebiegu wyjazdowego na szlak – *y2 1*
2. K2_ZNWj, zezwolenie na ustawienie przebiegu wjazdowego na stację – *y2 2*
3. K3_Vmax, zezwolenie na ustawienie sygnału zezwalającego na semaforze wyjazdowym – *y2 3*
4. /Pwl, odwzbudzenie przekaźnika Pwl (ciągłe) – *y2 13*

ii. Sygnały o stanie wewnętrznym blokady:

1. Pb, żarówka biała strzałki przyjazd (ciągłe lub migające) – *y2 4*
2. Pc, żarówka czerwona strzałki przyjazd (ciągłe lub migające) – *y2 5*
3. Ob, żarówka biała strzałki odjazd (ciągłe lub migające) – *y2 6*
4. Oc, żarówka czerwona strzałki odjazd (ciągłe) –y*2 7*
5. Kob, lampka biała Ko (ciągłe lub migające) – *y2 8*
6. Pwlc, lampka czerwona Pwl (ciągłe) – *y2 9*
7. dKob, lampka biała dKo (ciągłe) – *y2 10*
8. LdKo, licznik dKo (ciągłe) – *y2 11*
9. Dz, dzwonek (ciągłe) – *y2 12*

Powyżej w opisie wektorów oraz dalej w tabelach 4 i 5 zastosowano pewne oznaczenia użyte w albumach blokady [38][43] i urządzeń stacyjnych E [2][23] (system zależnościowy dla stacji kolejowych szeroko rozwijany w Polsce od lat 50’ XX wieku na licencji szwedzkiej firmy LM Ericsson) do opisu przekaźników pełniących określone funkcje:

Przekaźniki urządzeń E:

- S – sygnałowy (semafora wyjazdowego),
- Kc – kontrolny, kontroli światła czerwonego na semaforze (wjazdowym),

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- Kpz – kontrolny, kontroli świateł górnych (pomarańczowego lub zielonego) na semaforze (wjazdowym),
- Pml – pomocniczy liniowy,
- SSz – sygnałowy sygnału zastępczego (semafora wjazdowego),
- Jt – kontroli niezajętości odcinka torowego (pierwszego za semaforem wjazdowym),
- Jz - kontroli niezajętości odcinka zwrotnicowego (drugiego w kolejności odcinka za semaforem wjazdowym)

Przekaźniki blokady Eap [4][38][43]:

- Op – otrzymania pozwolenia,
- Dp – dania pozwolenia,
- Poz – pozwolenia dla sąsiedniego posterunku,
- Wbl – włączenia blokady,
- Zwbl – zwolnienia blokady,
- Nc – nadawania sygnału przemiennego,
- Pwl – przeciwwtórności liniowej,
- Zp – zwrotu pozwolenia,
- Lzs – liniowy zajętości szlaku,
- Jc – liniowy, odbierający sygnał przemienny,
- Od – wyjazdu z sąsiedniej stacji,
- oWbl – odwołania włączenia blokady,
- stKo – sterujący końcowy,
- dPo – doraźny początkowy,
- dKo – doraźny końcowy,
- W – włączający,
- Z – zwalniający,
- pS – powtarzający sygnałowy.

Sygnały opisano w tabeli 4 podając nazwy zmiennych wektorów, własne nazwy sygnałów, kierunki przekazywania informacji, opis słowny oraz określenie, czy przekazywana informacja jest krytyczna czy nie. Informacje krytyczne (polecenia, czyli informacje określające rodzaj działania przekazywane do sterowanego.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Tabela 4. Sygnały interfejsu. UP – układ pośredniczący, UK – urządzenia komputerowe. Źródło: opracowanie własne

Lp	Nazwa wektora	Numer zmiennej	Nazwa sygnału	Złożenie nazw	Skąd	Dokąd
1.	*2.*	*3.*	*4.*	*5.*	*6.*	*7.*
1.	x2	1	Op*Dp*/Wbl*Pwl	x2_1_Op*Dp*/Wbl*Pwl	Eap	UP
2.	x2	2	Op*Dp*/Wbl*pS	x2_2_Op*Dp*/Wbl*pS	Eap	UP
3.	x2	3	/Pwl	x2_3_/Pwl	Eap	UP
4.	x2	4	/W	x2_4_/W	Eap	UP
5.	x2	5	Przyjazd_biała	x2_5_Przyjazd_biała	Eap	UP
6.	x2	6	Przyjazd_czerwona	x2_6_Przyjazd_czerwona	Eap	UP
7.	x2	7	Odjazd_biała	x2_7_Odjazd_biała	Eap	UP
8.	x2	8	Odjazd_czerwona	x2_8_Odjazd_czerwona	Eap	UP
9.	x2	9	Lampka_Ko	x2_9_Lampka_Ko	Eap	UP
10.	x2	10	Lampka_Pwl	x2_10_Lampka_Pwl	Eap	UP
11.	x2	11	Lampka_dKo	x2_11_Lampka_dKo	Eap	UP
12.	x2	12	Licznik_dKo	x2_12_Licznik_dKo	Eap	UP
13.	x2	13	Licznik_dPo	x2_13_Licznik_dPo	Eap	UP
14.	x2	14	Dzwonek	x2_14_Dzwonek	Eap	UP
15.	y1	1	/Pml	y1_1_/Pml	UP	Eap
16.	y1	2	Pml	y1_2_Pml	UP	Eap
17.	y1	3	/S*/SSz*Kc*it	y1_3_/S*/SSz*Kc*it	UP	Eap
18.	y1	4	/it*/Jt*Jz	y1_4_/it*/Jt*Jz	UP	Eap
19.	y1	5	it*Jt*/Jz	y1_5_it*Jt*/Jz	UP	Eap
20.	y1	6	/Kc*Kpz	y1_6_/Kc*Kpz	UP	Eap
21.	y1	7	it	y1_7_it	UP	Eap
22.	y1	8	Przycisk_Wbl	y1_8_Przycisk_Wbl	UP	Eap
23.	y1	9	Przycisk_oWbl	y1_9_Przycisk_oWbl	UP	Eap
24.	y1	10	Przycisk_Poz	y1_10_Przycisk_Poz	UP	Eap
25.	y1	11	Przycisk_Ko	y1_11_Przycisk_Ko	UP	Eap
26.	y1	12	Przycisk_dKo	y1_12_Przycisk_dKo	UP	Eap
27.	y1	13	Przycisk_dPo	y1_13_Przycisk_dPo	UP	Eap
28.	x1	1	S1_Kc	x1_1_S1_Kc	UK	UP
29.	x1	2	S2_Kpz	x1_2_S2_Kpz	UK	UP
30.	x1	3	S3_/Sz	x1_3_S3_/Sz	UK	UP
31.	x1	4	S4_Jt	x1_4_S4_Jt	UK	UP
32.	x1	5	S5_Jz	x1_5_S5_Jz	UK	UP

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Lp	Nazwa wektora	Numer zmiennej	Nazwa sygnału	Złożenie nazw	Skąd	Dokąd
1.	*2.*	*3.*	*4.*	*5.*	*6.*	*7.*
33.	x1	6	S6_/Uwy	x1_6_S6_/Uwy	UK	UP
34.	x1	7	S7_it	x1_7_S7_it	UK	UP
35.	x1	8	DKO	x1_8_DKO	UK	UP
36.	x1	9	DPO	x1_9_DPO	UK	UP
37.	x1	10	KO	x1_10_KO	UK	UP
38.	x1	11	OWBL	x1_11_OWBL	UK	UP
39.	x1	12	PZK	x1_12_PZK	UK	UP
40.	x1	13	WBL	x1_13_WBL	UK	UP
41.	y2	1	K1_ZNWy	y2_1_K1_ZNWy	UP	UK
42.	y2	2	K2_ZNWj	y2_2_K2_ZNWj	UP	UK
43.	y2	3	K3_Vmax	y2_3_K3_Vmax	UP	UK
44.	y2	4	Pb	y2_4_Pb	UP	UK
45.	y2	5	Pc	y2_5_Pc	UP	UK
46.	y2	6	Ob	y2_6_Ob	UP	UK
47.	y2	7	Oc	y2_7_Oc	UP	UK
48.	y2	8	Kob	y2_8_Kob	UP	UK
49.	y2	9	Pwlc	y2_9_Pwlc	UP	UK
50.	y2	10	dKob	y2_10_dKob	UP	UK
51.	y2	11	LdKo	y2_11_LdKo	UP	UK
52.	y2	12	Dz	y2_12_Dz	UP	UK
53.	y2	13	/Pwl	y2_13_/Pwl	UP	UK

Zmienne wektorów **X1**, **X2**, **Y1** i **Y2** zostaną opisane funkcjami dla układu pośredniczącego.

Relacje pomiędzy sygnałami realizowane przez układ pośredniczący przedstawiono w tabeli 5.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Tabela 5. Relacje pomiędzy sygnałami. Źródło: opracowanie własne

Lp	Nazwa sygnału	=	Równanie
1.	*2.*	*3.*	*4.*
1.	y1_1_/Pml	=	/x1_6_S6_/Uwy
2.	y1_2_Pml	=	x1_6_S6_/Uwy
3.	y1_3_/S*/SSz*Kc*it	=	x1_1_S1_Kc * /x1_2_S2_Kpz * x1_3_S3_/Sz * x1_7_S7_it
4.	y1_4_/it*/Jt*Jz	=	/x1_7_S7_it * /x1_4_S4_Jt * x1_5_S5_Jz
5.	y1_5_it*Jt*/Jz	=	x1_7_S7_it * x1_4_S4_Jt * /x1_5_S5_Jz
6.	y1_6_/Kc*Kpz	=	/x1_1_S1_Kc * x1_2_S2_Kpz
7.	y1_7_it	=	x1_7_S7_it
8.	y1_8_Przycisk_Wbl	=	x1_13_WBL
9.	y1_9_Przycisk_oWbl	=	x1_11_OWBL
10.	y1_10_Przycisk_Poz	=	x1_12_PZK
11.	y1_11_Przycisk_Ko	=	x1_10_KO
12.	y1_12_Przycisk_dKo	=	x1_8_DKO
13.	y1_13_Przycisk_dPo	=	x1_9_DPO
14.	y2_1_K1_ZNWy	=	x2_1_Op*Dp*/Wbl*Pwl * x1_7_S7_it
15.	y2_2_K2_ZNWj	=	x2_4_/W
16.	y2_3_K3_Vmax	=	x2_2_Op*Dp*/Wbl*pS * x2_3_/Pwl * x1_6_S6_/Uwy
17.	y2_4_Pb	=	x2_5_Przyjazd_biała
18.	y2_5_Pc	=	x2_6_Przyjazd_czerwona
19.	y2_6_Ob	=	x2_7_Odjazd_biała
20.	y2_7_Oc	=	x2_8_Odjazd_czerwona
21.	y2_8_Kob	=	x2_9_Lampka_Ko
22.	y2_9_Pwlc	=	x2_10_Lampka_Pwl
23.	y2_10_dKob	=	x2_11_Lampka_dKo
24.	y2_11_LdKo	=	x2_12_Licznik_dKo
25.	y2_12_Dz	=	x2_14_Dzwonek
26.	y2_13_/Pwl	=	x2_3_/Pwl

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

6.4 Opis działania blokady

Opis słowny działania blokady znajduje się w artykule [4], albumie [43], skrypcie [23] i instrukcji [34]. Działanie blokady może zostać wyrażone także w postaci grafu przedstawionego na rysunku 22.

12. Ustawienie kierunku blokady z A do B
Nastawienie sygnału zezwalającego na semaforze wyjazdowym w A: PmI A
Odwołanie włączenia blokady w A: oWbI A
14. Sygnał zezwalający na semaforze wyjazdowym w A
Danie pozwolenia w B: Poz B
13. Zwrot pozwolenia przez A
Wyjazd pociągu z A na rozkaz lub Sz: dPo A
11. Inicjacja włączenia blokady przez A
Wyjazd pociągu z A: /PmI A
Odwołanie włączenia blokady w B: oWbI B
Włączenie blokady w A: WbI A
Odwołanie włączenia blokady w A: oWbI A
15. Pociąg na szlaku
Zwolnienie blokady w B: Ko B
1. Neutralny
Odwołanie włączenia blokady w A: oWbI A
4. Zwrot pozwolenia przez B
Zwolnienie blokady w A: Ko A
8. Pociąg na stacji A
Zwolnienie blokady w A: Ko A
Włączenie blokady w B: WbI B
Odwołanie włączenia blokady w B: oWbI B
Odwołanie włączenia blokady w B: oWbI B
10. Pociąg na stacji A po użyciu dKo
Wjazd na stację A: sekwencja it, Jt, Jz
Danie pozwolenia w A: Poz A
2. Inicjacja włączenia blokady przez B
3. Ustawienie kierunku blokady z B do A
7. Sygnał zezwalający na semaforze wjazdowym w A
Wjazd na stację A: sekwencja it, Jt, Jz
9. Oczekiwanie na wjazd pociągu na stację A
Wyjazd pociągu z B na rozkaz lub Sz: dPo B
Nastawienie sygnału zezwalającego na semaforze wyjazdowym w B: PmI B
Nastawienie sygnału zezwalającego na semaforze wjazdowym na stację A: /Kc, Kpz
Przygotowanie do wyjazdu pociągu na stację A na rozkaz lub Sz: dKo A
Wyjazd pociągu z B: /PmI B
6. Pociąg na szlaku
5. Sygnał zezwalający na semaforze wyjazdowym ze stacji B

Rysunek 22. Graf pracy blokady Eap z punktu widzenia obsługi poleceń z pulpitu i sygnałów wejściowych z otoczenia. Źródło: opracowanie własne na podstawie [34][38]

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Wierzchołki grafu reprezentują stany stabilne automatu, jaki realizuje blokada, z punktu widzenia jednego z dwóch zestawów współpracujących ze sobą do tego celu (tego znajdującego się na stacji A). Przejście z jednego stanu stabilnego do drugiego może odbyć się tylko wówczas, gdy operatorzy blokady (dyżurni ruchu na stacjach A i B) obsłużą określone polecenia (Wbl, Poz, oWbl, Ko, dKo, dPo wydawane bezpośrednio do blokady, albo pośrednio poprzez zmianę stanu urządzeń stacyjnych współpracujących z blokadą: Pml, Kc, Kpz) lub na blokadę oddziaływać będzie przejeżdżający pociąg poprzez zmianę stanu urządzeń kontroli niezajętości (it, Jz, Jt) i stacyjnych (Pml, Kc, Kpz). Na grafie polecenia operatorów i zmiany stanu urządzeń stacyjnych wpływających na zmianę stanu zostały opisane pogrubionymi skrótami. Pracownik może użyć przycisków na pulpicie, a pociąg oddziaływać na układy kontroli niezajętości elementów układu torowego (tory, rozjazdy). W poszczególnych łukach pomiędzy wierzchołkami grafu można by było wyróżnić inne stany stabilne. Przedstawienie każdej takiej zmiany któregokolwiek z przekaźników (np. kontroli niezajętości it, Jz, Jt) wpłynęło by negatywnie na czytelność grafu. Z tego powodu, a także ponieważ opis w postaci grafów nie jest jedynym możliwym, dla dokładniejszego wskazania czasów zmian stanów poszczególnych przekaźników wynikającego z ich konstrukcji i zastosowanych układów RC[12], możliwe jest zastosowanie harmonogramów ich pracy (w dziedzinie czasu) przy przechodzeniu pomiędzy poszczególnymi stanami stabilnymi.

Wykres kolejności pracy przekaźników [48], nazywany również harmonogramem pracy przekaźników [48] stanowi ujęcie kolejności pracy grupy przekaźników oraz ich wzajemnego uzależnienia w sposób graficzny.

W książce [48] w następujący sposób opisano zastosowaną symbolikę na takim wykresie. Na osi rzędnych oznaczone są poszczególne przekaźniki a na osi odciętych okresy pracy tych przekaźników. Długość poziomej linii wskazuje okres czasu pracy tych przekaźników (kotwica przyciągnięta, czynne styki zwarte). Linia wznosząca się z lewej strony do prawej oznacza czas przyciągania kotwicy od włączenia zasilania do chwili zwarcia styków czynnych spadająca zaś z lewej do prawej strony oznacza czas opadania kotwicy do chwili zwarcia styków biernych. Cienka linia pozioma oznacza okres czasu, w ciągu którego kotwica przekaźnika o opóźnionym działaniu pozostaje przyciągnięta po przerwaniu obwodu zasilania. Pionowe cienkie linie wskazują funkcjonalne powiązanie pracy poszczególnych przekaźników między sobą.

[12] Układ zawierający połączenie szeregowe kondensatora C i rezystora R, dołączany równolegle do cewki przekaźnika w celu wywołania opóźnienia odwzbudzenia tego przekaźnika.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Nieco inaczej wykres ten wygląda w źródłach późniejszych [6], choć nie została w nich określona zastosowana symbolika. Nie zmienia to faktu, że harmonogram również w sposób graficzny przedstawia kolejność pracy grupy przekaźników i ich wzajemne uzależnienie. Wykres jest wykonany w postaci tabeli, co sprawia, że jest czytelniejszy niż opisywany w [48]. Linie ukośne wznoszą się i opadają do krawędzi tabeli, co pozwala na zakreskowanie bądź zaczernienie prostokątów utworzonych na górnych podstawach trapezów. Części zakreskowane określają pracę przekaźnika wynikającą z zasilania cewki ze źródła zasilania a zaczernione odpowiadają stanowi czynnemu przekaźnika wynikającemu z jego opóźnienia na zwalnianie jako cechy konstrukcyjnej lub układu RC. Wpływ zmiany stanu jednego przekaźnika na inne zaznacza się pionowymi liniami przerywanymi. Strzałka oznacza w wierszu tabeli obsługę elementu sterowniczego przez personel sterowania ruchem (przycisku). Zastosowanie linii przerywanych poprawia czytelność.

W obu opisywanych reprezentacjach problematyczne wydaje się zastosowanie pionowych odcinków, które przechodząc przez wiele linii/wierszy odpowiadających przekaźnikom, których zmiana stanu może wynikać z innego faktu, utrudnia rozpoznanie co jest przyczyną a co skutkiem.

W książce [69] jest przedstawiony wykres o podobnych cechach, z tym że zrezygnowano z odcinków ukośnych natomiast linie pionowe zastąpiono ukośnymi strzałkami obrazującymi przyczynę i skutek. Grot strzałki wskazuje skutek. Jest to dość czytelne. Strzałki ukośne pozwalają dość dobrze wskazać co jest przyczyną a co skutkiem.

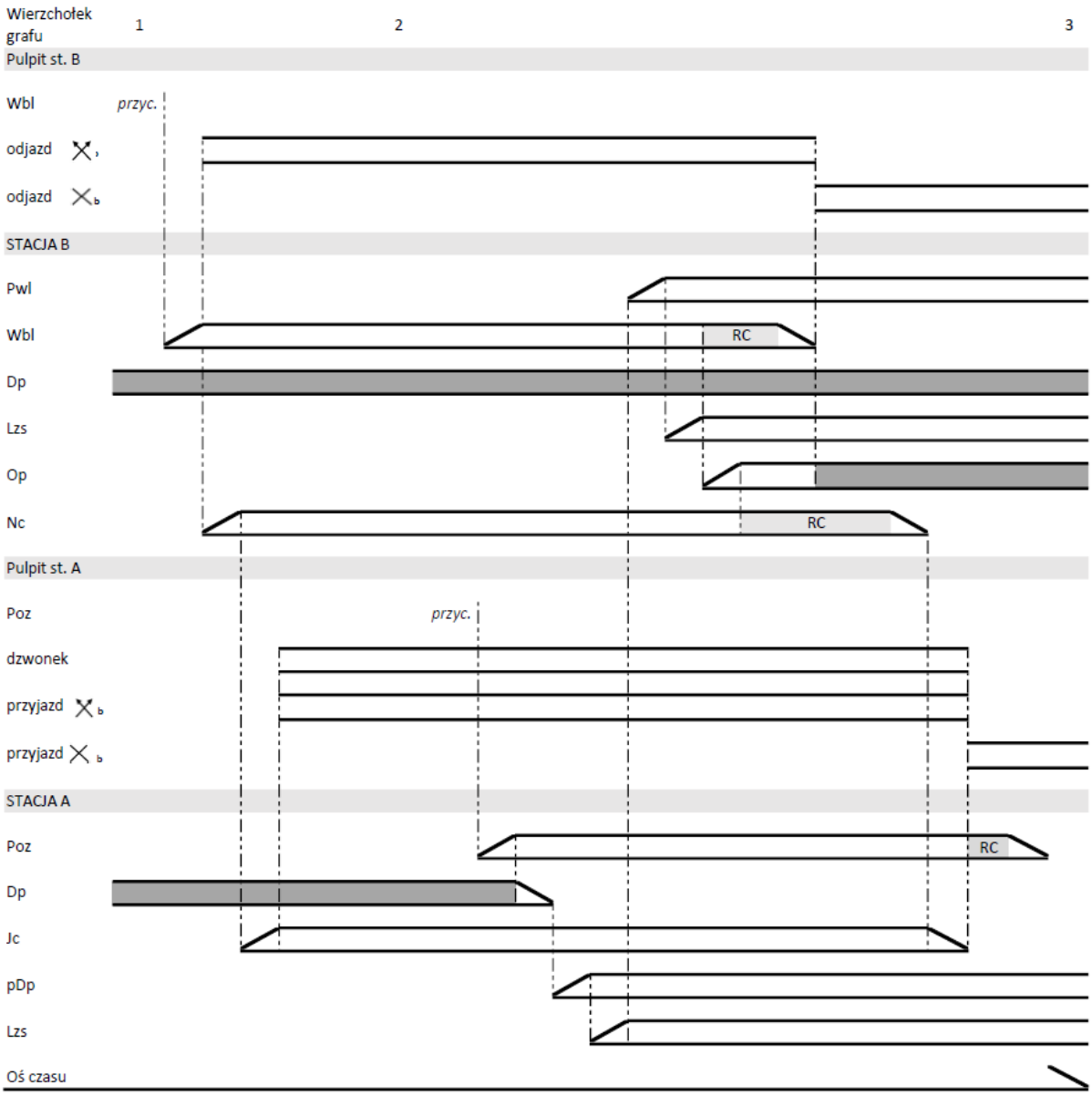
Cechy powyższych wykresów łączy wykres zastosowany w artykule [6], gdzie zastosowano trapezy i pionowe strzałki rysowane liniami przerywanymi. Pewnym nadmiarem jest jednak reprezentacja osobnych osi odciętych dla każdego przekaźnika obejmujących dziedzinę czasu. W tym zakresie czytelne i wystarczające wydaje się rozwiązanie tabelaryczne.

Na rysunku 23 przedstawiono przykład takiego harmonogramu dla blokady Eap. Legenda zastosowanych symboli znajduje się na rysunku 24.

Na osi rzędnych oznaczono poszczególne przekaźniki oraz wskazania na pulpitach, które biorą udział w danym przejściu. Na rysunku 23, na którym przedstawiono przejście ze stanu *1. Neutralnego* do stanu *3. Ustawienie kierunku blokady z B do A,* są to, patrząc od góry rysunku: odpowiednio dla stacji B przycisk Wbl, biała migająca lampka odjazd, ta sama lampka ale dla sytuacji świecenia światłem cągłym, przekaźniki Pwl, Wbl, Dp, Lzs, Op i Nc. Odpowiednio dla stacji A są to: przycisk Poz, dzwonek, biała lampka przyjazd migająca na

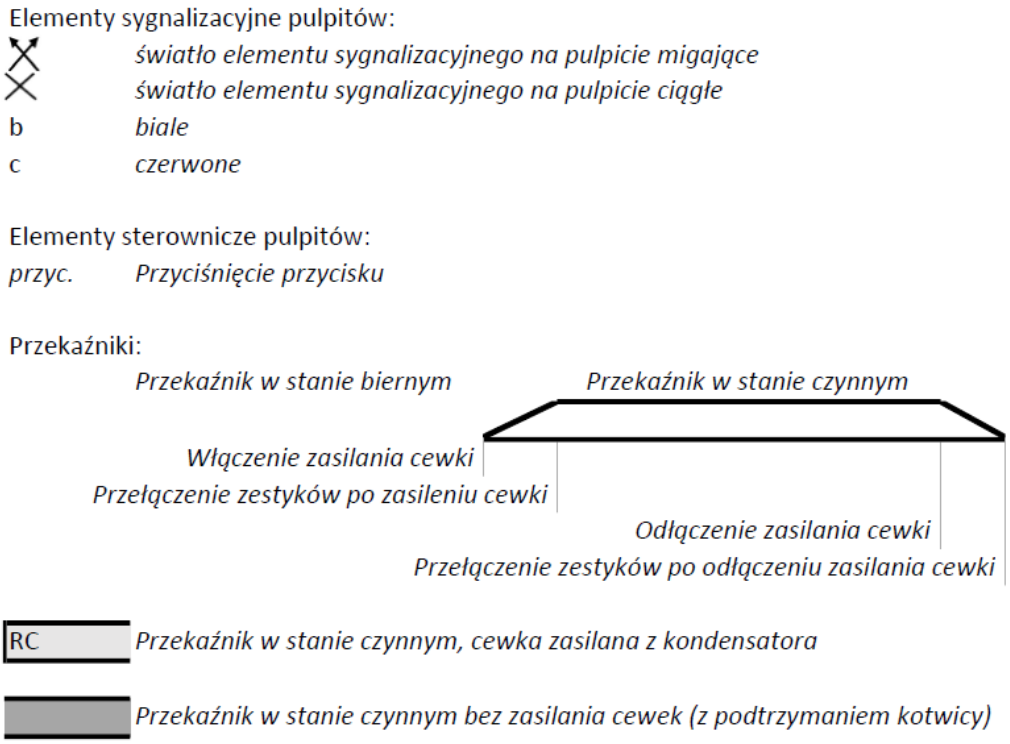
Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

biało i świecąca światłem białym ciągłym, przekaźniki Poz, Dp, Jc, pDp i Lzs. Oś odciętych, jest osią czasu. Obrazuje ona jak długo przekaźnik jest wzbudzony. Ukośna linia wznosząca się od lewej do prawej oznacza czas jaki jest potrzebny do wzbudzenia się przekaźnika – przyciągnięcie kotwicy, zwarcie styków czynnych. Opadająca linia przedstawia czas jaki jest potrzebny od odwzbudzenia się przekaźnika – opadnięcie kotwicy. Pionowe linie przerywane wskazują powiązania poszczególnych przekaźników ze sobą – jak zmiana stanu jednego z przekaźników oddziałuje na stan innego.



Rysunek 23. Harmonogram pracy przekaźników w blokadzie Eap podczas ustawiania kierunku ze stacji B do stacji A. Źródło: opracowanie własne na podstawie [38][8]

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21


Elementy sygnalizacyjne pulpitów:
światło elementu sygnalizacyjnego na pulpicie migające
światło elementu sygnalizacyjnego na pulpicie ciągłe
b biale
c czerwone
Elementy sterownicze pulpitów:
przyc. Przyciśnięcie przycisku
Przekaźniki:
Przekaźnik w stanie biernym
Przekaźnik w stanie czynnym
Włączenie zasilania cewki
Przełączenie zestyków po zasileniu cewki
Odłączenie zasilania cewki
Przełączenie zestyków po odłączeniu zasilania cewki
RC
Przekaźnik w stanie czynnym, cewka zasilana z kondensatora
Przekaźnik w stanie czynnym bez zasilania cewek (z podtrzymaniem kotwicy)

Rysunek 24. Legenda harmonogramu pracy przekaźników przedstawionego na rysunku 23. Źródło: opracowanie własne na podstawie [8]

Wskazane na rysunku przejście ze stanu *1.Neutralnego* do stanu *3. Ustawienie kierunku blokady z B do A* zapoczątkowane jest użyciem przycisku Wbl przez dyżurnego stacji B. Wciśnięcie przycisku Wbl na stacji B powoduje wzbudzenie się w znajdującym się tam zestawie blokady przekaźnika Wbl. Podtrzymuje się on na własnym zestyku bocznikującym zestyk przycisku na pulpicie (dyżurny może puścić przycisk). Spowoduje to w konsekwencji wzbudzenie się przekaźnika Nc. Przekaźnik Wbl ma swoje zestyki między innymi w obwodzie podświetlenia pulpitu na stacji B. Zostaje tam zwarty obwód strzałki ODJAZD, która miga na biało. Wzbudzony przekaźnik Nc wzbudza poprzez obwód liniowy przekaźnik Jc na stacji A. Wzbudzony przekaźnik Jc swoimi zestykami zwiera obwód podświetlenia pulpitu na stacji A. Spowoduje to miganie na biało strzałki PRZYJAZD oraz włączenie dzwonka. W ten sposób przedstawione zostało przejście do drugiego stanu stabilnego – *Inicjacja włączenia blokady przez B.*

Dalsze przejście do stanu *3. Ustawienie kierunku blokady z B do A* zapoczątkowane jest użyciem przycisku Poz przez dyżurnego stacji A. Powoduje ono wzbudzenie w zestawie znajdującym się na stacji A przekaźnika Poz. Podtrzymuje się on na własnym zestyku bocznikującym zestyk przycisku na pulpicie (dyżurny może puścić przycisk). Jego zestyk łączy obwód cewki odwzbudzającej przekaźnik Dp i przekaźnik ten się odwzbudza, co

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

w konsekwencji wzbudza kolejno przekaźnik pDp. Przekaźniki Dp i pDp zamykają obwód prądu stałego w linii transmisyjnej co wzbudza przekaźnik Lzs w zestawie stacji A i przekaźnik Pwl w zestawie stacji B. Wzbudzenie przekaźnika Pwl w zestawie na stacji B powoduje że znajdujący się tam przekaźnik Lzs wzbudza się. Wzbudzenie przekaźnika Lzs w zestawie na stacji B powoduje przerwanie zasilania cewki przekaźnika Wbl i przekaźnik ten (Wbl) odwzbudza się, przełączając obwody pulpitu w taki sposób, że strzałka kierunku ODJAZD przestaje migać a zaczyna świecić się światłem ciągłym. Inny zestyk przekaźnika Lzs w zestawie na stacji B łączy obwód zasilania cewki wzbudzającej przekaźnik Op i przekaźnik ten wzbudza się. Po wzbudzeniu przerywa on zasilanie cewki przekaźnika Nc i przekaźnik ten po pewnym czasie odwzbudza się przerywając nadawanie prądu przemiennego w linii transmisyjnej pomiędzy stacjami. To powoduje odwzbudzenie w zestawie na stacji A przekaźnika Jc. Odwzbudzenie przekaźnika Jc przełącza obwody pulpitu w taki sposób, że dzwonek milknie a strzałka PRZYJAZD przestaje migać a zaczyna świecić się światłem ciągłym. Inny zestyk przekaźnika Jc przerywa zasilanie cewki przekaźnika Poz i przekaźnik ten odwzbudza się. Został osiągnięty stan *3. Ustawienie kierunku blokady z B do A.*

Niektóre z przekaźników, takie jak Poz, Wbl, Zwbl, są opóźnione na zwalnianie. Posiadają rezystor i kondensator, który po wcześniejszym naładowaniu zasila cewkę przekaźnika, podtrzymując jego pracę wówczas, gdy jego obwód został przerwany. Na wykresach czasowych oznaczone to zostało za pomocą symbolu RC znajdującego się za pionową linią, oznaczającą przerwanie obwodu cewki danego przekaźnika.

Opis za pośrednictwem przedstawionego sposobu (harmonogramu pracy przekaźników) przejść pomiędzy wszystkimi stanami blokady Eap został przedstawiony w pracy [8].

Można skonstruować model blokady, na podstawie jej dokumentacji [38][43] i uwzględniwszy modele jej elementów opracowane w rozdziale 4.

W zakresie opisu urządzeń komputerowych zmiany stanów mogą zostać wskazane poprzez grafy, w postaci tabelarycznej przedstawiającej zmiany stanów wejść i wyjść oraz poprzez określenie czasu reakcji urządzeń komputerowych na zmiany stanów tych wejść i wyjść.

W ramach przykładu współpracy blokady Eap z urządzeniami komputerowymi wskazanego w rozdziale 6.2 zajęto się warstwą logiczną, gdzie możliwa jest operacja na sygnałach i grafach. Warstwa ta jest rzecz jasna związana z sygnałami elektrycznymi i czasem zmiany ich stanów.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

6.5 Specyfikacja interfejsu urządzeń komputerowych

6.5.1 Sygnały wejściowe do systemu komputerowego.

System komputerowy współpracujący z blokadą liniową przechowuje informację o dopuszczalnych kombinacjach sygnałów na wejściach. Wyspecyfikowane są kombinacje poprawnej pracy. W przypadku pozostałych kombinacji wejść, trwających dłużej niż 2 sekundy, następuje przejście do tzw. stanu bezpiecznego. Takie zachowanie jest odzwierciedleniem zasady przedstawionej w rozdziale 1.6.1.

Tabela 6. Przyporządkowanie określonym wartościom sygnałów elektrycznych sygnałów binarnych. Wejścia systemu komputerowego. Źródło – opracowanie własne na podstawie [33]

Lp.	Parametr	Miano	Wartość		Wartość logiczna
			minimalna	maksymalna	
1.	2.	3.	4.	5.	6.
1.	Napięcie	V DC	0	21,6	0
2.			21,6	36	1
3.			Pozostałe zakresy		Wejście uszkodzone

Powyższe przyporządkowanie można przedstawić także w postaci zależności (29), (30) i (31).

$$we_i^{log} = \text{"1"} \Leftrightarrow U_i \in \langle 21{,}6V; 36V \rangle \tag{29}$$

$$we_i^{log} = \text{"0"} \Leftrightarrow U_i \in \langle 0V; 21{,}6V) \tag{30}$$

$$we_i^{log} = \text{"}i - te\ wejście\ uszkodzone\text{"} \Leftrightarrow U_i \notin \langle 0V; 36V \rangle \tag{31}$$

gdzie:

we_i^{log} – stan logiczny *i*-tego wejścia,

U_i – wartość napięcia na *i*-tym wejściu,

i – numer wejścia, $i \in \{1, \dots, I\} \wedge i, I \in N^+$,

I – liczba wejść. Wektor **Y2** ma 13 składowych. $I = 13$.

W przypadku wejść urządzeń komputerowych, przy założeniu wykorzystania wprost sygnałów kierowanych z blokady Eap do pulpitu, które obejmują świecenie elementów pulpitu światłem ciągłym lub migającym lub brak świecenia, przyporządkowanie stanu blokady reprezentowanego normalnie wskazaniem pulpitu wymaga jeszcze przyporządkowania sygnałom binarnym wykrytym na określonych wejściach stanów logicznych elementów sygnalizacyjnych spośród następujących:

- ON – element świeci światłem ciągłym, wówczas gdy stan logiczny wynosi „1”,
- OFF – element nie świeci, wówczas gdy stan logiczny wynosi „0”

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- FLASH – element wyświetla sygnał migający, wówczas gdy stan logiczny wynosi naprzemiennie „0” i „1”,
- USZKODZONY – element uszkodzony, wówczas gdy stan logiczny wynosi „wejście uszkodzone”.

Do określenia prawidłowości migania światła (przemienne sygnały binarne) konieczne jest również określenie dopuszczalnej częstotliwości migania i wypełnienia impulsu. Wartości te w tym przypadku wynikają z działania układu migacza elektronicznego w blokadzie i na podstawie obserwacji można określić, że mieszczą się w zakresie od 1 do 2Hz. Stąd częstotliwość migania elementu $f_{mig} \in \langle 1Hz; 2Hz \rangle$.

Na tej podstawie, podobnie jak w rozdziale 1.6.2 dla kodera LEU, można określić w postaci zależności (32), (33) i (34) przyporządkowanie stanów logicznych stanom wejść urządzeń komputerowych.

$$we_i = \text{"ON"} \Leftrightarrow x_i(t) = 1 \wedge (x_i(t) = 1) \triangleq \left(we_i^{log} = \text{"1"}\right) \tag{32}$$

$$we_i = \text{"OFF"} \Leftrightarrow x_i(t) = 0 \wedge (x_i(t) = 0) \triangleq \left(we_i^{log} = \text{"0"}\right) \tag{33}$$

$$\begin{aligned} we_i = \text{"FLASH"} \Leftrightarrow x_i(t) = \frac{sgn(\sin 2\Pi f t) + 1}{2} \wedge f \in \langle 1Hz; 2Hz \rangle \wedge (x_i(t) = 1) \\ \triangleq \left(we_i^{log} = \text{"1"}\right) \wedge (x_i(t) = 0) \triangleq \left(we_i^{log} = \text{"0"}\right) \end{aligned} \tag{34}$$

Gdzie:

we_i - stan i-tego wejścia,

$x_i(t)$ - funkcja czasu sygnału logicznego i-tego wejścia.

Stanom wejść w określonej kombinacji odpowiada wykrycie określonego stanu blokady.

Przyporządkowanie stanów blokady do kombinacji stanów wejść wektora **Y2** urządzeń komputerowych na stacji A może zostać przedstawione w postaci tabeli prawdy (tabela 7).

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology

Tabela 7. Tabela prawdy - przyporządkowanie stanów blokady do kombinacji stanów wejść wektora **Y2** urządzeń komputerowych na stacji A. Źródło: opracowanie własne na podstawie [38]

Lp.	Stan nr j	Skrót opisujący stan nr j	Numer wejścia i	1	2	3	4	5	6	7	8	9	10	11	12	13	Stany poprzedzające (zbiór SP_j)
			Opis słowny wejścia i	K1_ZNWy	K2_ZNWj	K3_Vmax	Pb	Pc	Ob	Oc	Kob	Pwlc	dKob	LdKo	Dz	/Pwl	
			Opis słowny stanu j	Stan i-tego wejścia we_i													
1.	*2.*	*3.*	*4.*	*5.*	*6.*	*7.*	*8.*	*9.*	*10.*	*11.*	*12.*	*13.*	*14.*	*15.*	*16.*	*17.*	*18.*
1.	1	N	Neutralny (stan zasadniczy)	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	4;5;7;15;17;19;22
2.	2	BA	Blokada jest ustawiona ze stacji B do stacji A (na przyjazd)	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	4;22
3.	3	AB	Blokada jest ustawiona ze stacji A do stacji B (na odjazd)	ON	ON	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	5;22
4.	4	N-BA	Stan blokady był neutralny, stacja B poprosiła o pozwolenie	OFF	ON	OFF	FLASH	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	1;22
5.	5	N-AB	Stan blokady był neutralny, stacja A poprosiła o pozwolenie	OFF	ON	OFF	OFF	OFF	FLASH	OFF	OFF	OFF	OFF	OFF	OFF	ON	1;22
6.	6	BA-N3	Blokada była ustawiona ze stacji B do stacji A, stacja A zainicjowała zwolnienie blokady poleceniem KO (przejście do stanu N), faza 1	OFF	OFF	OFF	OFF	ON	OFF	OFF	FLASH	OFF	OFF	OFF	OFF	ON	14;22

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology

Lp.	Stan nr j	Skrót opisujący stan nr j	Numer wejścia i	1	2	3	4	5	6	7	8	9	10	11	12	13	Stany poprzedzające (zbiór SP_j)
			Opis słowny wejścia i	K1_ZNWy	K2_ZNWj	K3_Vmax	Pb	Pc	Ob	Oc	Kob	Pwlc	dKob	LdKo	Dz	/Pwl	
			Opis słowny stanu j	Stan i-tego wejścia we_i													
1.	*2.*	*3.*	*4.*	*5.*	*6.*	*7.*	*8.*	*9.*	*10.*	*11.*	*12.*	*13.*	*14.*	*15.*	*16.*	*17.*	*18.*
7.	7	AB-N3	Blokada była ustawiona ze stacji A do stacji B, stacja B zainicjowała zwolnienie blokady (przejście do stanu N)	OFF	ON	OFF	OFF	OFF	FLASH	OFF	OFF	OFF	OFF	OFF	ON	ON	11;22
8.	8	BA-Od	Blokada jest ustawiona ze stacji B do stacji A i w stanie, w którym jest właśnie wykorzystywane pozwolenie na wyjazd	OFF	ON	OFF	OFF	FLASH	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	2;22
9.	9	AB-Od	Blokada jest ustawiona ze stacji A do stacji B i w stanie, w którym jest właśnie wykorzystywane pozwolenie na wyjazd.	ON	ON	ON	OFF	OFF	ON	OFF	OFF	ON	OFF	OFF	OFF	ON	3;22
10.	10	BA-zaj	Blokada jest ustawiona ze stacji B do stacji A, przebieg wyjazdowy utracił kontrolę	OFF	ON	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	2;8;22

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology

Lp.	Stan nr j	Skrót opisujący stan nr j	Numer wejścia i	1	2	3	4	5	6	7	8	9	10	11	12	13	Stany poprzedzające (zbiór SP_j)
			Opis słowny wejścia i	K1_ZNWy	K2_ZNWj	K3_Vmax	Pb	Pc	Ob	Oc	Kob	Pwlc	dKob	LdKo	Dz	/Pwl	
			Opis słowny stanu j	Stan i-tego wejścia we_i													
1.	*2.*	*3.*	*4.*	*5.*	*6.*	*7.*	*8.*	*9.*	*10.*	*11.*	*12.*	*13.*	*14.*	*15.*	*16.*	*17.*	*18.*
			elementu lub operator wydał polecenie DPO w stacji B.														
11.	11	AB-zaj	Blokada jest ustawiona ze stacji A do stacji B, przebieg wyjazdowy utracił kontrolę elementu lub operator wydał polecenie DPO w stacji A	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	ON	3;9;22
12.	12	BA-dKo	Blokada jest ustawiona ze stacji B do stacji A, operator zlecił DKO w stacji A.	OFF	ON	OFF	OFF	ON	OFF	OFF	OFF	OFF	ON	ON	OFF	ON	10;22
13.	13	AB-N1	Blokada była ustawiona ze stacji A do stacji B, stacja A zainicjowała zwolnienie blokady (przejście do stanu N), faza 1	OFF	ON	OFF	OFF	OFF	FLASH	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3;22
14.	14	BA-WZ	Blokada jest ustawiona ze stacji B do stacji A, pociąg	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF	OFF	OFF	OFF	ON	10;22

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology

Lp.	Stan nr j	Skrót opisujący stan nr j	Numer wejścia i	1	2	3	4	5	6	7	8	9	10	11	12	13	Stany poprzedzające (zbiór SP_j)
			Opis słowny wejścia i	K1_ZNWy	K2_ZNWj	K3_Vmax	Pb	Pc	Ob	Oc	Kob	Pwlc	dKob	LdKo	Dz	/Pwl	
			Opis słowny stanu j	Stan i-tego wejścia we_i													
1.	*2.*	*3.*	*4.*	*5.*	*6.*	*7.*	*8.*	*9.*	*10.*	*11.*	*12.*	*13.*	*14.*	*15.*	*16.*	*17.*	*18.*
			opuścił szlak (wjechał na stację A)														
15.	15	AB-N2	Blokada była ustawiona ze stacji A do stacji B, stacja A zainicjowała zwolnienie blokady (przejście do stanu N), faza 2	OFF	ON	OFF	FLASH	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	13
16.	16	BA-N1	Blokada była ustawiona ze stacji B do stacji A, stacja B zainicjowała zwolnienie blokady (przejście do stanu N), faza 1	OFF	ON	OFF	FLASH	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	2
17.	17	BA-N2	Blokada była ustawiona ze stacji B do stacji A, stacja B zainicjowała zwolnienie blokady (przejście do stanu N), faza 2	OFF	ON	OFF	FLASH	OFF	OFF	OFF	FLASH	OFF	OFF	OFF	ON	ON	16;22
18.	18	BA-N4	Blokada była ustawiona ze stacji B do stacji	OFF	ON	OFF	OFF	OFF	FLASH	OFF	FLASH	OFF	OFF	OFF	ON	ON	6;21;22

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology

Lp.	Stan nr j	Skrót opisujący stan nr j	Numer wejścia i	1	2	3	4	5	6	7	8	9	10	11	12	13	Stany poprzedzające (zbiór SP_j)
			Opis słowny wejścia i	K1_ZNWy	K2_ZNWj	K3_Vmax	Pb	Pc	Ob	Oc	Kob	Pwlc	dKob	LdKo	Dz	/Pwl	
			Opis słowny stanu j	Stan i-tego wejścia we_i													
1.	*2.*	*3.*	*4.*	*5.*	*6.*	*7.*	*8.*	*9.*	*10.*	*11.*	*12.*	*13.*	*14.*	*15.*	*16.*	*17.*	*18.*
			A, stacja A zainicjowała zwolnienie blokady poleceniem KO (przejście do stanu N), faza 2														
19.	19	BA-N5	Blokada była ustawiona ze stacji B do stacji A, stacja A zainicjowała zwolnienie blokady poleceniem KO (przejście do stanu N), faza 3	OFF	ON	OFF	OFF	OFF	OFF	OFF	FLASH	OFF	OFF	OFF	OFF	ON	18;22
20.	20	BA-WZ2	Blokada jest ustawiona ze stacji B do stacji A, pociąg opuścił szlak (wjechał na stację A po DKO)	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF	ON	ON	OFF	ON	12;22
21.	21	BA-N6	Blokada była ustawiona ze stacji B do stacji A, stacja A zainicjowała zwolnienie blokady poleceniem KO	OFF	OFF	OFF	OFF	ON	OFF	OFF	FLASH	OFF	ON	ON	OFF	ON	20;22

Lp.	Stan nr j	Skrót opisujący stan nr j	Numer wejścia i	1	2	3	4	5	6	7	8	9	10	11	12	13	Stany poprzedzające (zbiór SP_j)
			Opis słowny wejścia i	K1_ZNWy	K2_ZNWj	K3_Vmax	Pb	Pc	Ob	Oc	Kob	Pwlc	dKob	LdKo	Dz	/Pwl	
			Opis słowny stanu j	Stan i-tego wejścia we_i													
1.	*2.*	*3.*	*4.*	*5.*	*6.*	*7.*	*8.*	*9.*	*10.*	*11.*	*12.*	*13.*	*14.*	*15.*	*16.*	*17.*	*18.*
			po DKO (przejście do stanu N), faza 1														

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology

Objaśnienie symboli zastosowanych w tabeli 7:

j – numer stanu blokady, $j \in$ J,

J – zbiór numerów stanów blokady, J $= \{1, \dots, 22\}$.

Powyższą tabelę można przedstawić w postaci macierzy $\boldsymbol{WE} = [we_{ji}]$. Jest to macierz prostokątna (dwuwymiarowa) o stałych wartościach elementów i stałych wymiarach 21 x 13. Każdy wiersz tabeli można potraktować jak wektor.

Dla każdego wiersza tabeli można określić równanie:

k^t – numer stanu blokady wykrywanego przez interfejs urządzeń komputerowych w chwili czasu t,

$$k^t = j \Leftrightarrow \text{w danej chwili } t \; \forall_{i \in I} we_i = we_{ji} \wedge k^{t-\tau} \epsilon SP_j \tag{35}$$

Gdzie:

SP_j – zbiór stanów mogących poprzedzać stan j. $SP_j \subset$ J,

$k^{t-\tau}$ – numer stanu blokady wykrywanego przez interfejs urządzeń komputerowych w chwili czasu $t - \tau$; $k^t, k^{t-\tau} \in$ J.

$t - \tau, t$ to kolejne chwile czasu.

Jeżeli wykrywany stan nie należy do wyspecyfikowanych, lub nie jest znany stan poprzedni (po uruchomieniu urządzeń komputerowych po przerwie ich pracy lub resecie), wówczas jego numer będzie wynosił 22.

$$k^t \notin J/\{22\} \vee k^{t-\tau} \in \emptyset \Rightarrow k^t = 22 \tag{36}$$

W stanie 22, po wykryciu stanu wejść odpowiadającego jednemu ze stanów z tabeli 7 następuje przejście do tego stanu. Istnieje wyjątek, ponieważ stany 4, 15 i 16 posiadają takie same stany wejść. W tej sytuacji następuje przejście do stanu 4. W przypadku, gdyby w rzeczywistości był to stan 15 powinno nastąpić po krótkim czasie przejście do stanu 1. Istnieją łuki ze stanów 15 i 4 do stanu 1, więc można wybrać dowolny. Inaczej jest ze stanem 16, z którego przejście może być wyłącznie do stanu 17, po wydaniu polecenia OWBL na stacji A. Stąd graf jest uzupełniony o łuk ze stanu 4 do stanu 17, który zostanie wykorzystany w tej sytuacji i po wydaniu polecenia OWBL.

Wejściowe sygnały zależnościowe K1_ZNWy, K2_ZNWj i K3_Vmax do urządzeń stacyjnych (komputerowych, znajdujących się na stacji A przykładu, wskazanej na rysunku 20) mogą zostać opisane słownie w następujący sposób.

Zasilone wejście K1_ZNWy wskazuje, że droga przebiegu umożliwiająca wyjazd ze stacji na szlak może być nastawiona i utwierdzona przez urządzenia stacyjne.

Wejście K1_ZNWy powinno zostać zasilone, jeśli spełnione są łącznie następujące warunki:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- Pozwolenie na wyjazd jest wydawane dla stacji A (blokada liniowa jest w stanie AB),
- Odcinek ochronny[13] stacji A (JtZ) jest wolny,
- Żaden pociąg nie znajduje się na szlaku (itAB wolny),
- Odcinek ochronny stacji B (JtA) jest wolny[14].

Zasilone wejście K2_ZNWj wskazuje, że może być ustawiony przez urządzenia stacyjne sygnał zezwalający na wjazd na stację (na semaforze Z).

Zasilone wejście K3_Vmax wskazuje, że na sygnalizatorze wyjazdowym (K, L albo M) może być wyświetlony sygnał zezwalający na wyjazd.

Wejście K3_Vmax powinno zostać zasilone, jeżeli spełnione są łącznie następujące warunki:

- Pozwolenie na wyjazd jest wydawane dla stacji A (blokada liniowa jest w stanie AB),
- Odcinek ochronny stacji A (JtZ) jest wolny,
- Żaden pociąg nie znajduje się na szlaku (itAB wolny),
- Odcinek ochronny stacji B (JtA) jest wolny[15],
- Droga przebiegu wyjazdowego ze stacji A została nastawiona i utwierdzona a jej elementy są kontrolowane.

Wejście /Pwl powinno zostać zasilone gdy przekaźnik Pwl nie jest wzbudzony.

Blokada Eap dostarcza sygnałów o swoim stanie w celu przekazania operatorowi pulpitu, w tym o przyjęciu lub odrzuceniu poleceń operatora. W niniejszym przykładzie założono, że sygnały te są tożsame z sygnałami dostarczanymi do rzeczywistego pulpitu. Możliwe, i stosowane, są inne rozwiązania polegające na odczycie stanu poszczególnych przekaźników blokady. Wymagają one jednak albo skonstruowania zestawu w taki sposób, aby mógł on współpracować z urządzeniami komputerowymi, albo wykonania modyfikacji w istniejącym zestawie podczas dostosowywania go do powiązania w miejscu jego instalacji. Ta ostatnia metoda wymaga m.in. przerywania i łączenia na nowo obwodów wewnątrz zestawu, a zatem istotnej modyfikacji blokady.

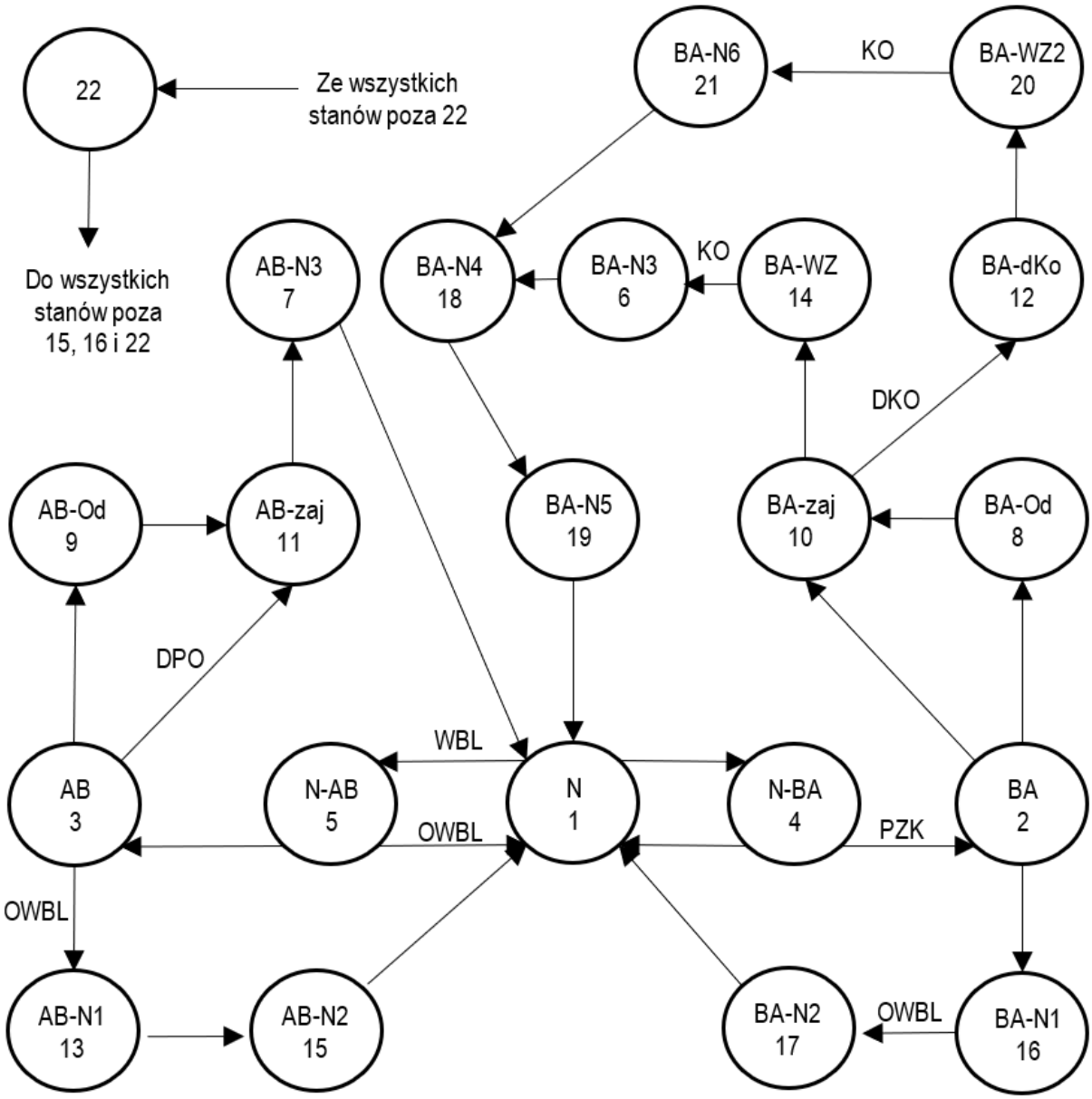
Warstwa logiczna zawiera również opis dopuszczalnych przejść pomiędzy stanami np. w postaci grafu (rysunek 25) lub tabeli przejść.

[13] Przepisy dotyczące projektowania rozmieszczenia sygnalizatorów przytorowych [79] wymagają zastosowania odcinka o długości minimum 100m za semaforem wjazdowym. Zgodnie z teorią związana z przebiegiem [19] jest to droga ochronna, chociaż odcinek ten nie jest w przepisach [79] tak nazywany. Zastosowano wobec tego określenie „odcinek ochronny". W odniesieniu do przykładu wskazanego na rysunku 20 odcinki ochronne wchodzą w skład odcinków kontroli niezajętości JtZ i JtA.

[14] W praktyce odcinek itAB projektuje się tak, aby obejmował sobą odcinki ochronne na obu stacjach.

[15] Jak wyżej.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21



Rysunek 25. Graf stanów komputerowych urządzeń stacyjnych dla przypadku współpracy z blokadą Eap. Źródło: opracowanie własne na podstawie [38] i obserwacji pulpitów stanowiska laboratoryjnego

Opisy ponad łukami grafu na rysunku 25 dotyczą oczekiwanej reakcji na wydane polecenia (na stacji A).

6.5.2 Sygnały wyjściowe z systemu komputerowego.

System komputerowy realizuje funkcje zależnościowe i umożliwia wpływanie operatora na blokadę Eap poprzez polecenia wydawane za pośrednictwem pulpitu komputerowego. Ich efektem są sygnały wyjściowe.

Sygnały przekazywane są poprzez bezpotencjałowe zestyki przekaźników. Dostępne są zestyki zwierne i rozwierne. Zwarcie zestyku można rozważać jako zmniejszenie oporu elektrycznego

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

tego elementu poniżej pewnej wartości. Rozwarcie zestyku można rozważać jako zwiększenie oporu elektrycznego tego elementu powyżej pewnej wartości. Można to ująć w postaci tabeli 8.

Tabela 8. Przyporządkowanie określonym wartościom oporu elektrycznego zestyków sygnałów binarnych. Wyjścia systemu komputerowego. Źródło – opracowanie własne na podstawie [51]

Lp.	Rodzaj zestyku	Wartość oporu		Wartość logiczna sygnału wyjściowego
		minimalna	maksymalna	
1.	*2.*	*3.*	*4.*	*5.*
1.	Zwierny	0	0,03Ω	1
2.		3MΩ	∞	0
3.		Pozostałe zakresy		Nie można określić - zestyk uszkodzony
4.	Rozwierny	0	0,03Ω	0
5.		3MΩ	∞	1
6.		Pozostałe zakresy		Nie można określić - zestyk uszkodzony

Powyższe przyporządkowanie można przedstawić także w postaci zależności (37) i (38).

$$\forall_{zz_l \in LZZ_l} \forall_{zr_l \in LZR_l} R_l^{zz_l} \in \langle 0; 0{,}03\Omega \rangle \wedge R_l^{zr_l} \in \langle 3M\Omega; \infty) \Rightarrow wy_l^{log} = \text{"1"} \quad (37)$$

$$\forall_{zz_l \in LZZ_l} \forall_{zr_l \in LZR_l} R_l^{zr_l} \in \langle 0; 0{,}03\Omega \rangle \wedge R_l^{zz_l} \in \langle 3M\Omega; \infty) \Rightarrow wy_l^{log} = \text{"0"} \quad (38)$$

gdzie:

l – numer wyjścia, $l \in \{1, \dots, L\} \wedge l, L \in N^+$,

L – liczba wyjść, Wektor $\mathbf{X1}$ ma 13 składowych $L = 13$,

wy_l^{log} – stan logiczny *l*-tego wyjścia,

zz_l – numer zestyku zwiernego *l*-tego wyjścia, $zz_l \in LZZ_l$

LZZ_l – zbiór numerów zestyków zwiernych *l*-tego wyjścia, $LZZ_l \in N^+$

$R_l^{zz_l}$ – wartość oporu na *zz*-tym zestyku zwiernym *l*-tego wyjścia, $R_l^{zz_l} \in R^+ \cup \{0\}$

zr_l – numer zestyku rozwiernego *l*-tego wyjścia, $zr_l \in LZR_l$

LZR_l – zbiór numerów zestyków rozwiernych *l*-tego wyjścia, $LZR_l \in N^+$

$R_l^{zr_l}$ – wartość oporu na *zr*-tym zestyku rozwiernym *l*-tego wyjścia, $R_l^{zr_l} \in R^+ \cup \{0\}$

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Polecenia

Wiadomym jest, że blokada odrzuci polecenia, których nie będzie można w danym jej stanie zrealizować. Jednocześnie pozwala to na założenie, że stan blokady wykrywany przez system komputerowy może być zafałszowany usterką, która wystąpiła w połączeniach pomiędzy urządzeniami.

Jeżeli system komputerowy przekaże do blokady polecenie operatora (wydane przez niego do blokady za pomocą pulpitu urządzeń stacyjnych) blokada Eap zareaguje na to polecenie poprzez zmianę stanu określoną łukiem opisanym skrótem polecenia na grafie na rysunku 25. Jeżeli odpowiedź ta nie zostanie odebrana przez urządzenia stacyjne w ciągu 2 sekundy, urządzenia te wskazują operatorowi, że polecenie zostało odrzucone.

Polecenia operatora powinny powodować zmianę wyjścia na czas 2 sekund, co wynika z zapisów instrukcji Ie-10 [31] (§22 ust. 1 zdanie drugie: „*Przyciski i dźwigienki niestabilne należy obsługiwać zdecydowanie, utrzymując je w zmienionym położeniu około 2 sekundy.*„).

Skróty poleceń zostały zapisane wielkimi literami:

- DKO – doraźny początkowy, wy_8^{log},
- DPO – doraźny końcowy, wy_9^{log},
- KO – końcowy, wy_{10}^{log},
- OWBL – odwołanie włączenia blokady, wy_{11}^{log},
- PZK – dania pozwolenia, wy_{12}^{log},
- WBL – włączenia blokady, wy_{13}^{log}.

Powyższe polecenia zostają przerwane po wykryciu niezgodnej kombinacji sygnałów wejściowych K1_ZNWy i K3_Vmax.

Ponieważ wcześniej określono, że przy wystąpieniu kombinacji sygnałów wejściowych innej niż określona w tabeli 7 dłużej niż 2 sekundy następuje przejście do stanu 22, przerwanie poleceń należy zapisać za pośrednictwem relacji (39).

$$k^t = 22 \wedge \left(\overline{we_1^{log}} \cdot we_3^{log} = 1\right) \Rightarrow \bigwedge_{l=8}^{13} wy_l^{log} := \text{"0"} \tag{39}$$

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Sygnały o stanie urządzeń stacyjnych

O stanie urządzeń stacyjnych świadczą sygnały, które są związane z faktem realizacji zależności w urządzeniach stacyjnych, kontroli stanu urządzeń zewnętrznych (semafora wjazdowego, odcinków kontroli niezajętości) i poleceń operatora tych urządzeń związanych z nastawianiem drogi przebiegu i jej rozwiązywaniem, nastawianiem sygnałów na semaforach. Zależności te są na tyle skomplikowane, że zostaną przedstawione w postaci osobnych grafów i funkcji. Wynika to również z faktu, że interfejs jest opisywany z wykorzystaniem wejść i wyjść i związanych z nimi sygnałów. Sygnałów wyjściowych poza tymi dotyczącymi poleceń do blokady, opisanymi w poprzednim punkcie „Polecenia", jest 7:

- S1_Kc – światło czerwone na semaforze wjazdowym, wy_1^{log},
- S2_Kpz – sygnał zezwalający (inny niż zastępczy) na semaforze wjazdowym, wy_2^{log},
- S3_/Sz – nie wyświetlony sygnał zastępczy na semaforze wjazdowym, wy_3^{log},
- S4_Jt – pierwszy odcinek kontroli niezajętości za semaforem wjazdowym wolny, wy_4^{log},
- S5_Jz – drugi odcinek kontroli niezajętości za semaforem wjazdowym wolny, wy_5^{log},
- S6_/Uwy – utwierdzona droga przebiegu wyjazdowego, wy_6^{log},
- S7_it – odcinek pokrywający szlak wolny, wy_7^{log}.

W celu oznaczenia zależności nie mających odzwierciedlenia w wejściach i wyjściach interfejsu, ani w zależnościach czasowych, wprowadzono zmienne zew_m. Gdzie: m – numer zmiennej obcej względem sygnałów interfejsu. $m \in N^+$. W szczególności dana zmienna oznaczać będzie właściwy stan urządzeń zależnościowych.

zew_1 - spełnione zależności i polecenia operatora kontrolowane w obwodzie przekaźnika sygnałowego oprócz we_1^{log} i we_3^{log}, wy_4^{log}, wy_5^{log} i wy_6^{log}.

zew_2 - odwzbudzony przekaźnik utwierdzający przebiegu wjazdowego,

zew_3 - sprawna żarówka główna komory światła czerwonego semafora wjazdowego,

zew_4 - sprawne żarówki kombinacji komór świateł zielonych, pomarańczowych i wskaźników W24, W19, W20 adekwatnych do nastawionej drogi przebiegu wjazdowego,

zew_5 - polecenie operatora do nastawienia sygnału zastępczego (Sz) na semaforze wjazdowym,

zew_6 - polecenie operatora do wyłączenia sygnału zastępczego (Sz) na semaforze wjazdowym.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

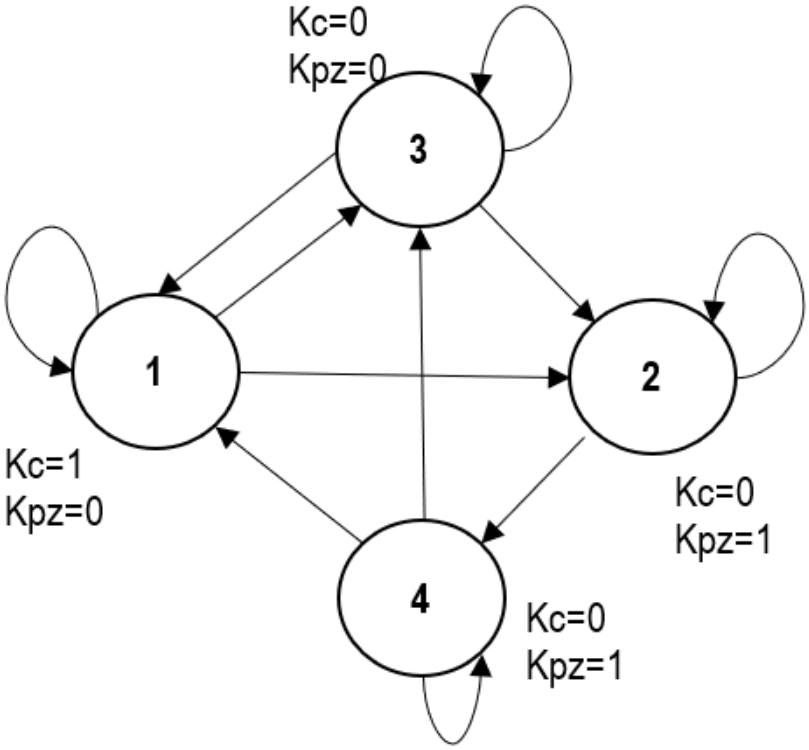
wy_1^{log} (S1_Kc) i wy_2^{log} (S2_Kpz) są to dwa wyjścia uzależnione od wskazań semafora wjazdowego na stację. Te wskazania zależą natomiast od stanu elementów drogi przebiegu i decyzji operatora urządzeń.

Graf dla sygnałów Kc (wy_1^{log}) i Kpz (wy_2^{log}) semafora wjazdowego przedstawiony został na rysunku 26. Symbolem s^t oznaczono numer stanu w chwili czasu t.

$$s^t = 1 \Rightarrow \left(wy_1^{log} = 1\right) \wedge \left(wy_2^{log} = 0\right)$$

$$s^t \in \{2,4\} \Rightarrow \left(wy_1^{log} = 0\right) \wedge \left(wy_2^{log} = 1\right)$$

$$s^t = 3 \Rightarrow \left(wy_1^{log} = 0\right) \wedge \left(wy_2^{log} = 0\right)$$



Rysunek 26. Graf dla sygnałów Kc i Kpz semafora wjazdowego. Źródło: opracowanie własne, na podstawie [2]

Występowanie stanu 4 jest związane z faktem opóźnienia wprowadzonego przy odwzbudzaniu przekaźnika Kpz i wzbudzaniu przekaźnika Kc. Jest to zabieg, który powinien być celowo zastosowany w urządzeniach stacyjnych dla semafora wjazdowego współpracującego z blokadą półsamoczynną Eap. Powyższy graf uwzględnia podstawowe zakłócenie jakim jest przepalenie żarówki.

Warunki przejścia dla krawędzi grafu są sformułowane w następujący sposób (poniżej, podobnie jak w rozdziale 4.5, użyto symbolu ->, w celu wskazania przejścia ze stanu poprzedniego o numerze $s^{t-\tau}$ (w poprzedniej chwili czasu $t-\tau$) do stanu następnego o numerze s^t (w aktualnej chwili czasu t)):

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

(1->2) $(s^{t-\tau}=1)\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=1\right)\Rightarrow s^t$ $:=2$

(2->2) $(s^{t-\tau}=2)\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=1\right)\Rightarrow s^t$ $:=2$

(2->4) $(s^{t-\tau}=2)\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=0\right)\Rightarrow s^t$ $:=4\ \wedge t_1:=t$

Gdzie:

t_1 – zmienna pomocnicza czasu, związana z odmierzaniem go dla odwzbudzenia przekaźnika Kpz i wzbudzenia Kc, $t_1\in T$

Przyjęto przy tym, że zmiana stanu przekaźników Kpz (odwzbudzenie) i Kc (wzbudzenie zajmuje 2s od momentu przerwania obwodu świateł zezwalających.

(4->4) $s^{t-\tau}=4\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=0\right)\wedge t<$ $t_1+2s\Rightarrow s^t:=4$

(4->1) $s^{t-\tau}=4\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=0\right)\wedge t\geq$ $t_1+2s\wedge zew_3=1\Rightarrow s^t:=1\wedge t_1:=0$

(1->1) $s^{t-\tau}=1\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=0\right)\wedge$ $zew_3=1\Rightarrow s^t:=1$

(4->3) $s^{t-\tau}=4\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=0\right)\wedge t\geq$ $t_1+2s\wedge zew_3=0\Rightarrow s^t:=3\wedge t_1:=0$

(3->3) $s^{t-\tau}=3\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=0\right)\wedge$ $zew_3=0\Rightarrow s^t:=3$

(3->2) $s^{t-\tau}=3\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=1\right)\Rightarrow s^t$ $:=2$

(3->1) $s^{t-\tau}=3\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=0\right)\wedge$ $zew_3=1\Rightarrow s^t:=1$

(1->3) $s^{t-\tau}=1\wedge\left(zew_1\cdot zew_2\cdot zew_4\cdot we_1^{log}\cdot we_3^{log}\cdot wy_4^{log}\cdot wy_5^{log}\cdot\overline{wy_6^{log}}=0\right)\wedge$ $zew_3=0\Rightarrow s^t:=3$

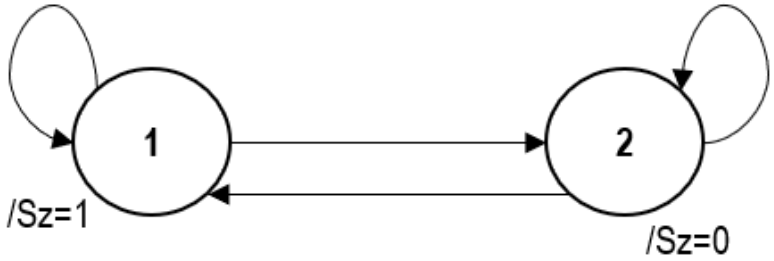
Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

wy_3^{log} (S3_/Sz) to wyjście uzależnione od polecenia nastawienia lub wyłączenia sygnału zastępczego przez personel i układu odmierzającego czas trwania próby podawania takiego sygnału na semaforze. Czas ten jest ograniczony do 90s, co wynika z zapisów instrukcji Ie-12 (E24) § 23 ust. 3 pkt. 6) [30]. W aktualnym stanie przepisów polecenie sygnału zastępczego nie jest uwarunkowane żadnymi zależnościami poza czasem.

$$s^t = 1 \Rightarrow wy_3^{log} = 1$$

$$s^t = 2 \Rightarrow wy_3^{log} = 0$$

Graf dla sygnału S3_/Sz (wy_3^{log}) semafora wjazdowego przedstawiony został na rysunku 27.



Rysunek 27. Graf dla sygnału S3_/Sz. Źródło: opracowanie własne

Warunki przejścia dla krawędzi grafu są sformułowane w następujący sposób:

(1->1) $s^{t-\tau} = 1 \wedge zew_5 = 0 \Rightarrow s^t := 1$

(1->2) $s^{t-\tau} = 1 \wedge zew_5 = 1 \wedge zew_6 = 0 \Rightarrow s^t := 2 \wedge t_2 := t$

(2->2) $s^{t-\tau} = 2 \wedge zew_6 = 0 \wedge t < t_2 + 90s \Rightarrow s^t := 2$

(2->1) $s^{t-\tau} = 2 \wedge (zew_6 = 1 \vee t \geq t_2 + 90s) \Rightarrow s^t := 1 \wedge t_2 := 0$

Gdzie:

t_2 – zmienna pomocnicza czasu, związana z odmierzaniem go dla samoczynnego wyłączenia sygnału zastępczego, $t_2 \in T$

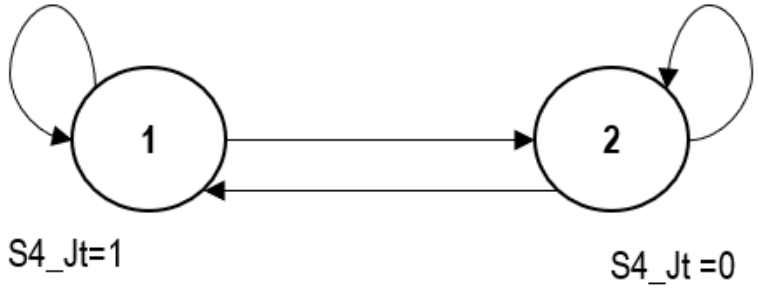
S4_Jt, S5_Jz i S7_it to wyjścia nie uzależnione wprost od stanu urządzeń stacyjnych i blokady. Ich stan zależy od obecności pociągu na danym odcinku kontroli niezajętości, na które został podzielony układ torowy stacji i szlaku:

- Stan S4_Jt (wy_4^{log}) zależy od obecności pociągu na odcinku Jt (0) lub braku tej obecności (1). Odcinek Jt to odcinek znajdujący się za semaforem wjazdowym – pierwszy w kolejności podczas wjazdu na stację.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

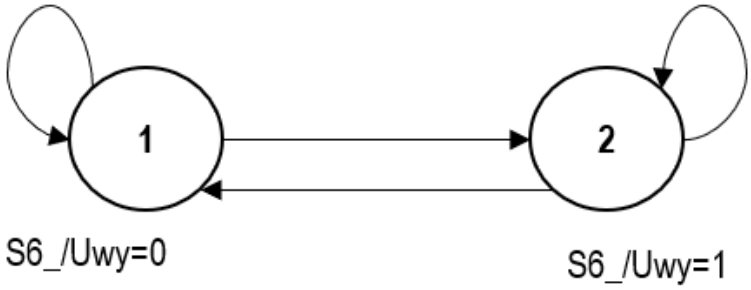
- Stan S5_Jz (wy_5^{log}) zależy od obecności pociągu na odcinku Jz (0) lub braku tej obecności (1). Odcinek Jz to odcinek znajdujący się za semaforem wjazdowym – drugi w kolejności podczas wjazdu na stację.
- Stan S7_it (wy_7^{log}) zależy od obecności pociągu na odcinku it (0) lub braku tej obecności (1). Odcinek it to odcinek znajdujący się pomiędzy semaforami wjazdowymi stacji połączonych rozważaną blokadą. Pokrywa on szlak.

Grafy dla sygnałów S4_Jt, S5_Jz i S7_it są do siebie podobne. W związku z tym na rysunku przedstawiono jeden dla sygnału S4_Jt.



Rysunek 28. Graf dla sygnału S4_Jt. Źródło: opracowanie własne

wy_6^{log} (S6_/Uwy) to wyjście uzależnione od stanu utwierdzenia drogi przebiegu wyjazdowego i spełnionych zależności związanych z utrzymywaniem sygnału zezwalającego na semaforze wyjazdowym. Graf dla tego sygnału został przedstawiony na rysunku 29.



Rysunek 29. Graf dla sygnału S6_/Uwy. Źródło: opracowanie własne

$$s^t = 1 \Rightarrow wy_6^{log} = 0$$

$$s^t = 2 \Rightarrow wy_6^{log} = 1$$

Warunki przejścia dla krawędzi grafu są sformułowane w następujący sposób:

(1->1) $s^{t-\tau} = 1 \wedge \left(zew_1 \cdot zew_2 \cdot we_1^{log} \cdot we_3^{log} \cdot wy_4^{log} \cdot wy_5^{log} = 0\right) \Rightarrow s^t := 1$

(1->2) $s^{t-\tau} = 1 \wedge \left(zew_1 \cdot zew_2 \cdot we_1^{log} \cdot we_3^{log} \cdot wy_4^{log} \cdot wy_5^{log} = 1\right) \Rightarrow s^t := 2$

(2->2) $s^{t-\tau} = 2 \wedge \left(zew_1 \cdot zew_2 \cdot we_1^{log} \cdot we_3^{log} \cdot wy_4^{log} \cdot wy_5^{log} = 1\right) \Rightarrow s^t := 2$

(2->1) $s^{t-\tau} = 2 \wedge \left(zew_1 \cdot zew_2 \cdot we_1^{log} \cdot we_3^{log} \cdot wy_4^{log} \cdot wy_5^{log} = 0\right) \Rightarrow s^t := 1$

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

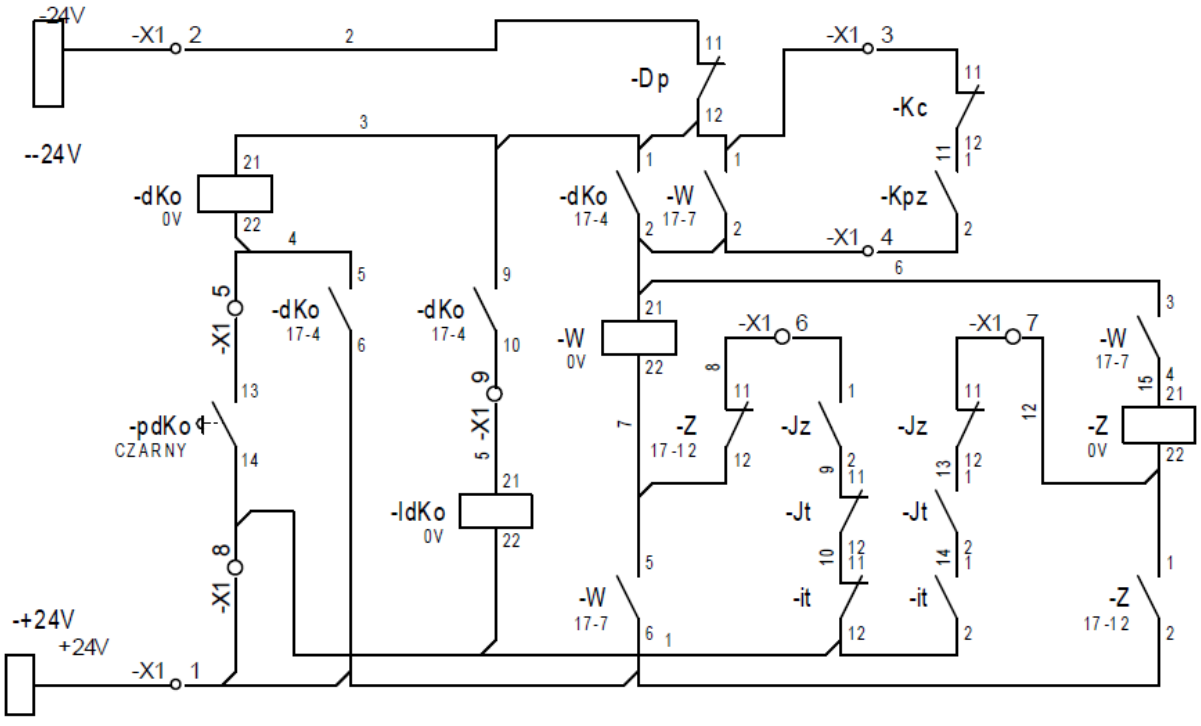
6.6 Badanie wybranego układu i jego powiązania

6.6.1 Wybór układu

Z uwagi na skomplikowanie całego powiązania wytypowano fragment obejmujący tzw. obwód oddziaływania pociągów na blokadę. Obwód ten składa się z cewek trzech przekaźników (W – włączającego, Z – zwalniającego, i dKo – doraźnego końcowego), ich zestyków oraz cewki licznika ldKo, przycisku pdKo (znajdujących się na pulpicie nastawczym, a zatem w ramach powiązania z komputerowymi urządzeniami na stacji, obsługiwanymi poprzez pulpit komputerowy) i zestyków przekaźników blokady (Dp – dania pozwolenia) i stacyjnych urządzeń komputerowych (Kc, Kpz, it, Jt, Jz).

6.6.2 Opracowanie schematu, zestawień i modelu

Opracowano opisywany schemat na podstawie arkusza zawartego w dokumentacji blokady Eap [38] i poprzednich doświadczeń z obwodem zmodyfikowanym [35]. Arkusz zawiera połączenia cewek, zestyków i przycisków. Arkusz przedstawiony został na rysunku 30.



Rysunek 30. Schemat połączeń obwodu oddziaływania pociągów na blokadę. Źródło: opracowanie własne na podstawie [35][38]

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Obwód na rysunku 30 został narysowany w programie SEE Electrical Expert [41][42]. Program ten umożliwia wygenerowanie listy wykorzystanych aparatów oraz zestawienia połączeń pomiędzy ich wyprowadzeniami.

Tabela 9. Lista aparatów badanego układu. Źródło: opracowanie własne na podstawie pliku wygenerowanego przez program SEE Electrical Expert

Etykieta symbolu	**Opis produktu**	**Kod materiałowy**	**Producent**	**Ilość: Kod materiałowy**
dKo	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1
Dp	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, bistabilny, dwuuzw	przekaźnik_do_badan_ remanencyjny		1
it	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1
Jt	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1
Jz	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1
Kc	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1
Kpz	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1
ldKo	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

pdKo	PRZYCISK Z ZESTYKAMI 1ZZ 5A 300V	1SFA619100R1011	ABB	1
W	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1
X1	ZŁĄCZKA ŚRUBOWA V0.UL 94 4mm2	039001	LEGRAND	1
Z	Przekaźnik elektromagnetyczny na prąd stały, napięciowy, obojętny, jednouzw	Przekaźnik_do_badan		1

Do celów eksperymentów opracowano przekaźniki: obojętny (kod materiałowy: Przekaźnik_do_badan) i bistabilny (przekaźnik_do_badan_remanencyjny). Każdy zawiera po pięć zestyków zwiernych i pięć rozwiernych, co jest wystarczające do modelowania układu.

Tabela 10. Zestawienie połączeń dla badanego układu. Źródło: opracowanie własne na podstawie pliku wygenerowanego przez program SEE Electrical Expert

Numer potencjału	**Początek**	**Koniec**
1	+-X1:1	+24V:
1	+-it:2	+-it:12
1	+-W:6	+-Z:2
1	+-X1:1	+-X1:8
1	+-it:12	+-X1:8
1	+-X1:8	+-ldKo:22
1	+-pdKo:14	+-X1:8
1	+-dKo:6	+-W:6
1	+-dKo:6	+-X1:1
2	+-Dp:11	+-X1:2
2	+-X1:2	-24V:
3	+-dKo:9	+-dKo:1
3	+-dKo:9	+-dKo:21
3	+-W:1	+-Dp:12
3	+-dKo:1	+-Dp:12
3	+-Kc:11	+-X1:3
3	+-W:1	+-X1:3
4	+-pdKo:13	+-X1:5
4	+-dKo:5	+-X1:5
4	+-dKo:5	+-dKo:22
5	+-X1:9	+-ldKo:21
5	+-dKo:10	+-X1:9

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

6	+-W:3	+-W:21
6	+-Kpz:2	+-X1:4
6	+-W:2	+-X1:4
6	+-dKo:2	+-W:21
6	+-W:2	+-dKo:2
7	+-Z:12	+-W:5
7	+-W:5	+-W:22
8	+-Jz:1	+-X1:6
8	+-Z:11	+-X1:6
9	+-Jz:2	+-Jt:11
10	+-Jt:12	+-it:11
11	+-Kc:12	+-Kpz:1
12	+-Z:22	+-Z:1
12	+-Z:22	+-X1:7
12	+-Jz:11	+-X1:7
13	+-Jt:1	+-Jz:12
14	+-Jt:2	+-it:1
15	+-W:4	+-Z:21

Na podstawie tych danych został wygenerowany model – plik tekstowy zawierający odwołania do modeli ogólnych elementów oraz przedstawiający ich nazwy i logiczne związki pomiędzy nimi. Plik został przedstawiony w załączniku numer 1 do rozprawy.

Model ten posłużył do dalszych badań w środowisku Dedan. Na podstawie modelu ogólnego został opracowany model dokładniejszy z wykorzystaniem makrogeneratora (uwzgledniający m.in. właściwe liczby stanów aparatów użytych w modelu).

6.6.3 Określenie oczekiwanego działania

Określenie oczekiwanego działania urządzenia (i jego modelu) w określonym stanie i przy określonych wymuszeniach opiera się o wprowadzenie grafu środowiska Env na podstawie opisu słownego. Opis słowny może dotyczyć zdarzeń prostych: np. wzbudzenia lub odwzbudzenia przekaźnika (lub ich grupy) ale wynikających z zasad związanych z funkcjonowanie badanego urządzenia (wymagań). Dla blokady półsamoczynnej można przedstawić wymagania, które obejmą również jej powiązanie z urządzeniami stacyjnymi, sformułowane m.in. w instrukcji Ie-4 [79]:

„4. Zwolnienie blokady powinno następować po spełnieniu warunków zwolnienia:

a) zajęcie urządzenia odziaływania za semaforem wjazdowym,

b) wyświetlenie sygnału „Stój” na semaforze wjazdowym,

c) zwolnienie szlakowego urządzenia oddziaływania,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

d) zwolnienie urządzenia odziaływania za semaforem wjazdowym.

(...)

5. W przypadku wjazdu na sygnał zastępczy (Sz) na semaforze wjazdowym, wszystkie warunki zwolnienia blokady powinny zostać spełnione i skontrolowane przez system blokady."

Badany układ blokady Eap i jego powiązanie z urządzeniami stacyjnymi (np. komputerowymi) powinien pozwolić na realizowanie powyższych zasad z tym, że:

- Przejeżdżający pociąg oddziałuje na odcinki kontroli niezajętości pokrywające układ torowy a stan tych odcinków jest przekazywany do blokady za pośrednictwem zestyków sterowanych z urządzeń stacyjnych.
- Wyświetlenie sygnału „Stój" na semaforze wjazdowym jest uwarunkowane zajętością odcinków Jt lub Jz. Zależność ta jest realizowana wewnątrz urządzeń stacyjnych.

Prawidłowe działanie układu powinno być następujące:

1. W stanie neutralnym:
   1.1. odcinki Jt, Jz i iz są wolne (przekaźniki Jt, Jz i it są wzbudzone),
   1.2. semafor wjazdowy wskazuje sygnał „Stój" – przekaźnik Kc wzbudzony a Kpz odwzbudzony,
   1.3. blokada jest w stanie neutralnym – przekaźnik Dp wzbudzony bezprądowo.
   1.4. Przycisk pdKo nie zwiera obwodu.
   1.5. Przekaźnik dKo jest odwzbudzony.
   1.6. Cewka ldKo nie jest zasilona.
2. Jazda ze stacji na szlak:
   2.1. Blokada zostaje ustawiona na wyjazd – nie zmienia to stanu przekaźników w badanym układzie,
   2.2. Pociąg wyjeżdża na szlak w jednej z możliwych sekwencji zajmowania i zwalniania odcinków i związanej z tym zmiany stanu przekaźników Jt, Jz i iz.
   2.3. Pozostałe przekaźniki (poza Jt, Jz i iz) nie zmieniają stanu.
3. Jazda do stacji ze szlaku, bez usterek:
   3.1. Blokada zostaje ustawiona na wjazd – zmienia to stan przekaźnika Dp, zostaje on odwzbudzony i pozostaje w tym stanie.
   3.2. Pociąg jedzie ze szlaku na stację w jednej z możliwych sekwencji zajmowania i zwalniania odcinków i związanej z tym zmiany stanu przekaźników Jt, Jz i iz.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

3.3. W trakcie sekwencji jazdy, o której mowa w punkcie 3.2, zostaje nastawiony sygnał zezwalający na semaforze wjazdowym – zawsze przed zajętością odcinka Jt. Może zostać zatem ustawiony przed zajętością iz lub już w trakcie tej zajętości. Nastawienie sygnału zezwalającego oznacza zmianę stanu przekaźników Kc i Kpz: wpierw Kpz zostaje wzbudzony a dopiero potem Kc odwzbudzony.

3.4. W trakcie sekwencji jazdy, o której mowa w punkcie 3.2 zostaje nastawiony sygnał „Stój" – zawsze z chwilą zajęcia odcinka Jt. Nastawienie sygnału „Stój" oznacza zmianę stanu przekaźników Kc i Kpz: wpierw Kpz zostaje odwzbudzony a dopiero potem Kc wzbudzony. W sekwencji omawianej w tym punkcie istotne jest celowe wprowadzenie opóźnienia omawianej zmiany stanu Kc i Kpz. Pomiędzy odwzbudzeniem Kpz a wzbudzeniem Kc upływa dość krótki czas, natomiast dłuższy jest pomiędzy zajęciem odcinka Jt (odwzbudzeniem przekaźnika Jt) a odwzbudzeniem Kpz. W tym wydłużonym czasie ma zmienić stan przekaźnik W (ten czas powinien na to pozwolić).

3.5. W trakcie sekwencji jazdy, o której mowa w punkcie 3.2 przekaźnik W wzbudza się – zawsze po zajęciu odcinka Jt i pozostaje wzbudzony.

3.6. W trakcie sekwencji jazdy, o której mowa w punkcie 3.2 przekaźnik Z wzbudza się – zawsze po zwolnieniu odcinka Jt i pozostaje wzbudzony.

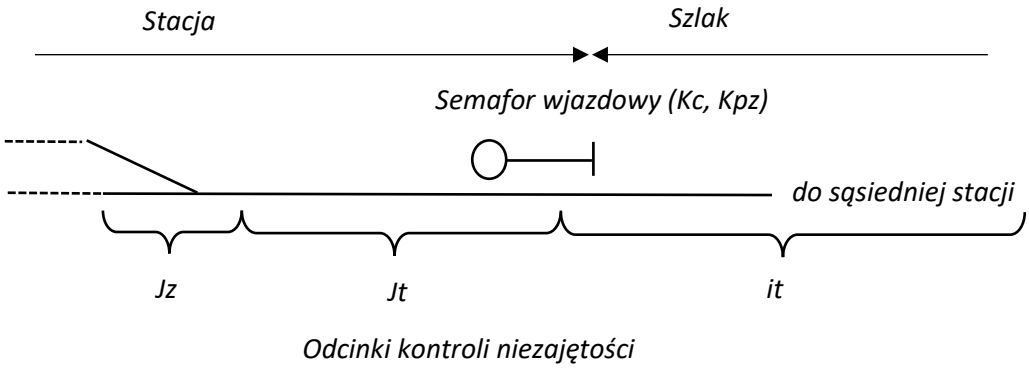
3.7. Wzbudzone przekaźniki W i Z dają możliwość przywrócenia blokadzie stanu neutralnego, czyli zmiany stanu przekaźnika Dp. Po wzbudzeniu Dp przekaźnik ten pozostaje wzbudzony. Wzbudzenie przekaźnika Dp przerywa obwody cewek W i Z i przekaźniki te odwzbudzają się. Zmiana stanu Dp, W i Z może nastąpić wobec tego tylko gdy Jt jest wolny ale zarówno gdy pociąg zajmuje jeszcze Jz lub gdy zakończył realizację sekwencji wjazdu i Jz jest już wolne.

Prawidłową jest jazda, w której pojazd szynowy, który co do zasady musi posiadać przynajmniej dwie osie, przejeżdżając z jednego odcinka kontroli niezajętości na drugi zajmuje na pewien czas oba te odcinki.

Również przy tym nie „gubi" nic po drodze, a zatem zajmowanie i zwalnianie odcinków przebiega w określonej kolejności wynikającej z kierunku jazdy. Kolejność przedstawiono w tabeli 11. Pociąg może mieć różną długość. Wyróżniono pociąg „długi", tj. taki, który jest dłuższy niż odcinek Jt, oraz „krótki" – równy, bądź krótszy niż odcinek Jt.

Na rysunku 31 przedstawiono fragment układu torowego z oznaczeniem odcinków kontroli niezajętości.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21



Rysunek 31. Rozmieszczenie odcinków kontroli niezajętości it, Jt i Jz przy semaforze wjazdowym stacji. Źródło: opracowanie własne

Gdy dany odcinek jest wolny i układ kontroli niezajętości sprawny, to odpowiadający mu przekaźnik (it, Jt, Jz) jest wzbudzony (kotwica uniesiona do rdzenia cewki – up).

Gdy dany odcinek jest zajęty lub układ kontroli niezajętości niesprawny, to odpowiadający mu przekaźnik (it, Jt, Jz) jest odwzbudzony (kotwica nie uniesiona do rdzenia cewki – down).

Tabela 11. Prawidłowa kolejność zajmowania i zwalniania odcinków podczas jazdy pociągu. Źródło: opracowanie własne

Kolejność zdarzeń	**Stacja**	**Jz**	**Jt**	**it**	**Szlak**
1.	*2.*	*3.*	*4.*	*5.*	*6.*
1. Stan zasadniczy					
1.	Stacja	Wolny (up)	Wolny (up)	Wolny (up)	Szlak
2. Jazda do stacji (<-)					
2.1. Wariant 1 – pociąg długi					
1.	Stacja	Wolny (up)	Wolny (up)	Wolny (up)	Szlak
2.		Wolny (up)	Wolny (up)	Zajęty (down)	
3.		Wolny (up)	Zajęty (down)	Zajęty (down)	
4.		Zajęty (down)	Zajęty (down)	Zajęty (down)	
5.		Zajęty (down)	Zajęty (down)	Wolny (up)	
6.		Zajęty (down)	Wolny (up)	Wolny (up)	
7.		Wolny (up)	Wolny (up)	Wolny (up)	
2.2. Wariant 2 – pociąg krótki					
1.	Stacja	Wolny (up)	Wolny (up)	Wolny (up)	Szlak
2.		Wolny (up)	Wolny (up)	Zajęty (down)	
3.		Wolny (up)	Zajęty (down)	Zajęty (down)	
4.		Wolny (up)	Zajęty (down)	Wolny (up)	

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Kolejność zdarzeń	Stacja	Jz	Jt	it	Szlak
1.	*2.*	*3.*	*4.*	*5.*	*6.*
5.		Zajęty (down)	Zajęty (down)	Wolny (up)	
6.		Zajęty (down)	Wolny (up)	Wolny (up)	
7.		Wolny (up)	Wolny (up)	Wolny (up)	
2.3. Wariant 3 (mało prawdopodobny) – pociąg długości odcinka Jt					
1.	Stacja	Wolny (up)	Wolny (up)	Wolny (up)	Szlak
2.		Wolny (up)	Wolny (up)	Zajęty (down)	
3.		Wolny (up)	Zajęty (down)	Zajęty (down)	
4.		Zajęty (down)	Zajęty (down)	Wolny (up)	
5.		Zajęty (down)	Wolny (up)	Wolny (up)	
6.		Wolny (up)	Wolny (up)	Wolny (up)	
3. Jazda ze stacji (->)					
3.1. Wariant 1 – pociąg długi					
1.	Stacja	Wolny (up)	Wolny (up)	Wolny (up)	Szlak
2.		Zajęty (down)	Wolny (up)	Wolny (up)	
3.		Zajęty (down)	Zajęty (down)	Wolny (up)	
4.		Zajęty (down)	Zajęty (down)	Zajęty (down)	
5.		Wolny (up)	Zajęty (down)	Zajęty (down)	
6.		Wolny (up)	Wolny (up)	Zajęty (down)	
7.		Wolny (up)	Wolny (up)	Wolny (up)	
3.2. Wariant 2 – pociąg krótki					
1.	Stacja	Wolny (up)	Wolny (up)	Wolny (up)	Szlak
2.		Zajęty (down)	Wolny (up)	Wolny (up)	
3.		Zajęty (down)	Zajęty (down)	Wolny (up)	
4.		Wolny (up)	Zajęty (down)	Wolny (up)	
5.		Wolny (up)	Zajęty (down)	Zajęty (down)	
6.		Wolny (up)	Wolny (up)	Zajęty (down)	
7.		Wolny (up)	Wolny (up)	Wolny (up)	
3.3. Wariant 3 (mało prawdopodobny) – pociąg długości odcinka Jt					
1.	Stacja	Wolny (up)	Wolny (up)	Wolny (up)	Szlak

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Kolejność zdarzeń	Stacja	Jz	Jt	it	Szlak
1.	*2.*	*3.*	*4.*	*5.*	*6.*
2.		Zajęty (down)	Wolny (up)	Wolny (up)	
3.		Zajęty (down)	Zajęty (down)	Wolny (up)	
4.		Wolny (up)	Zajęty (down)	Zajęty (down)	
5.		Wolny (up)	Wolny (up)	Zajęty (down)	
6.		Wolny (up)	Wolny (up)	Wolny (up)	

6.6.4 Uzupełnienie modelu o oczekiwane działanie

Określenie oczekiwanego działania urządzenia w określonym stanie i przy określonych wymuszeniach zostało wprowadzone do modelu w postaci opisu serwera środowiska Env na podstawie opisu słownego zawartego w rozdziale 6.6.3.

Przykładowy opis serwera środowiska dla jazdy do stacji pociągu najdłuższego, przy braku usterek i w oparciu o blokadę wygląda w następujący sposób (po znakach `//` w danej linijce znajduje się komentarz nie uwzględniany dalej przez program):

```
server: Env (servers ArmZ,ArmW,cDp,cV,
     sw1_Dp,sw2_Dp,it,Jt,Jz,Kpz,Kc;
     agents AEnv,APZ,APW)
states {start,V,DpG,
     Dp_on1,Dp_on2,Dp_on1a,Dp_of1,Dp_of2,
     it_of,it_on,Jt_of,Jt_on,Jt_ofW,Jt_onZ,
     Kpz_on,Kpz_of,Kc_on,Kc_of,Jz_on,Jz_of,
     wait_WZ,ofW,ofZ,ofBoth}
}
services {ini,ok,sWON,sZON,sWOF,sZOF,wait}
actions {
{AEnv.Env.ini, Env.start} -> {AEnv.cDp.g_1[1], Env.DpG},
{AEnv.Env.ok, Env.DpG} -> {AEnv.cV.v_1[1], Env.V},
//krok 1a Dp of
{AEnv.Env.ok, Env.V} -> {AEnv.sw2_Dp.on, Env.Dp_of1},
// ale potem wycofać, bo Dp działa dwustabilnie
{AEnv.Env.ok, Env.Dp_of1} -> {AEnv.sw2_Dp.of, Env.Dp_of2},
//krok 2 it of
{AEnv.Env.ok, Env.Dp_of2} -> {AEnv.it.of, Env.it_of},
//krok 2a Kpz on
{AEnv.Env.ok, Env.it_of} -> {AEnv.Kpz.on, Env.Kpz_on},
//krok 2b Kc of
{AEnv.Env.ok, Env.Kpz_on} -> {AEnv.Kc.of, Env.Kc_of},
//krok 3 Jt of
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
{AEnv.Env.ok, Env.Kc_of} -> {AEnv.Jt.of, Env.Jt_of},
//krok 3a - modelowanie opóźnienia - czekanie na wzbudzenie W
{APW.Env.sWON, Env.Jt_of} -> {APZ.ArmW.ok, Env.Jt_ofW},
//krok 3b Kpz of
{AEnv.Env.ok, Env.Jt_ofW} -> {AEnv.Kpz.of, Env.Kpz_of},
//krok 3c Kc on
{AEnv.Env.ok, Env.Kpz_of} -> {AEnv.Kc.on, Env.Kc_on},
//krok 4 Jz of
{AEnv.Env.ok, Env.Kc_on} -> {AEnv.Jz.of, Env.Jz_of},
//krok 5 it on
{AEnv.Env.ok, Env.Jz_of} -> {AEnv.it.on, Env.it_on},
//krok 6 Jt on
{AEnv.Env.ok, Env.it_on} -> {AEnv.Jt.on, Env.Jt_on},
//krok 6a - modelowanie opóźnienia - czekanie na wzbudzenie Z
{APZ.Env.sZON, Env.Jt_on} -> {APZ.ArmZ.ok, Env.Jt_onZ},
//krok 6b - odwzbudzenie Dp (niedeterministycznie)
{AEnv.Env.ok, Env.Jt_onZ} -> {AEnv.sw1_Dp.on, Env.Dp_on1},
// ale potem wyofać bo Dp działa dwustabilnie
{AEnv.Env.ok, Env.Dp_on1} -> {AEnv.sw1_Dp.of, Env.Dp_on2},
//krok 7
{AEnv.Env.ok, Env.Jt_onZ} -> {AEnv.Jz.on, Env.Jz_on},
//krok 7a - teraz Dp już musi się odwzbudzić
{AEnv.Env.ok, Env.Jz_on} -> {AEnv.sw1_Dp.on, Env.Dp_on1a},
// ale potem wycofać bo Dp działa dwustabilnie
{AEnv.Env.ok, Env.Dp_on1a} -> {AEnv.sw1_Dp.of, Env.Dp_on2},
//sytuacja pośrednia - początek impulsu na Dp w stanie 6
//ale koniec w stanie 7
{AEnv.Env.ok, Env.Dp_on1} -> {AEnv.Jz.on, Env.Dp_on1a},
{AEnv.Env.ok, Env.Dp_on2} -> {AEnv.Env.wait, Env.wait_WZ},
//czekanie na odwzbudzenie W i Z
{APZ.Env.sZOF, Env.wait_WZ} -> {Env.ofZ},
{APW.Env.sWOF, Env.wait_WZ} -> {Env.ofW},
{APZ.Env.sZOF, Env.ofW} -> {Env.ofBoth},
{APW.Env.sWOF, Env.ofZ} -> {Env.ofBoth},
//terminacja AEnv
{AEnv.Env.wait, Env.ofBoth} -> {Env.ofBoth},
};
```

6.6.5 Wyniki eksperymentów

Przeprowadzono 7 eksperymentów:

- przy jeździe pociągu do stacji, przy braku usterek obwodu i normalnej pracy blokady (obsługa zgodnie z instrukcją [34], jazda pociągu na sygnał zezwalający inny niż zastępczy [32]):
    1. dla pociągu krótkiego,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

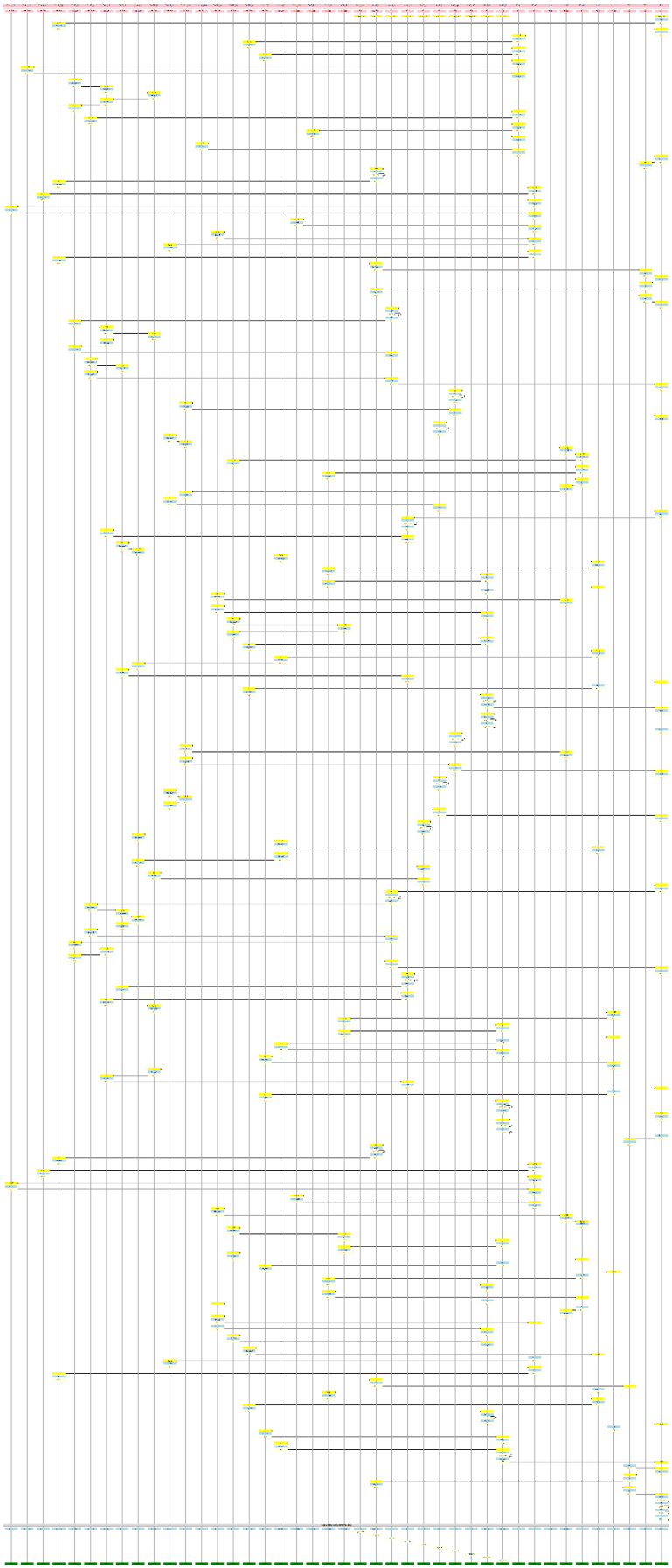
2. dla pociągu długiego,

- przy jeździe pociągu ze stacji, przy braku usterek obwodu i normalnej pracy blokady (obsługa zgodnie z instrukcją [34], jazda pociągu na sygnał zezwalający inny niż zastępczy [32]):
    3. dla pociągu krótkiego,
    4. dla pociągu długiego,
- przy jeździe do stacji pociągu długiego i wybranych usterkach:
    5. przerwie w cewce przekaźnika W w dowolnej chwili (obsługa zgodnie z instrukcją [34], jazda pociągu na sygnał zezwalający inny niż zastępczy [32]),
    6. przerwie w cewce przekaźnika W po wzbudzeniu (obsługa zgodnie z instrukcją [34], jazda pociągu na sygnał zezwalający inny niż zastępczy [32]),
    7. przerwie w cewce przekaźnika W na końcu pracy układu, po odwzbudzeniu tego przekaźnika (obsługa zgodnie z instrukcją [34], jazda pociągu na sygnał zezwalający inny niż zastępczy [32]).

Eksperymenty przeprowadzone bez usterek urządzeń (1-4) zakończyły się pomyślnie: żaden z serwerów reprezentujących elementy modelowanego układu się nie zakleszczył, a wszyscy kluczowi agenci ulegli terminacji. Inni agenci pozostają w stanie „technicznego" zakleszczenia, to znaczy oczekują w swoich komunikatach „neutralnych" w kotwiach przekaźników.

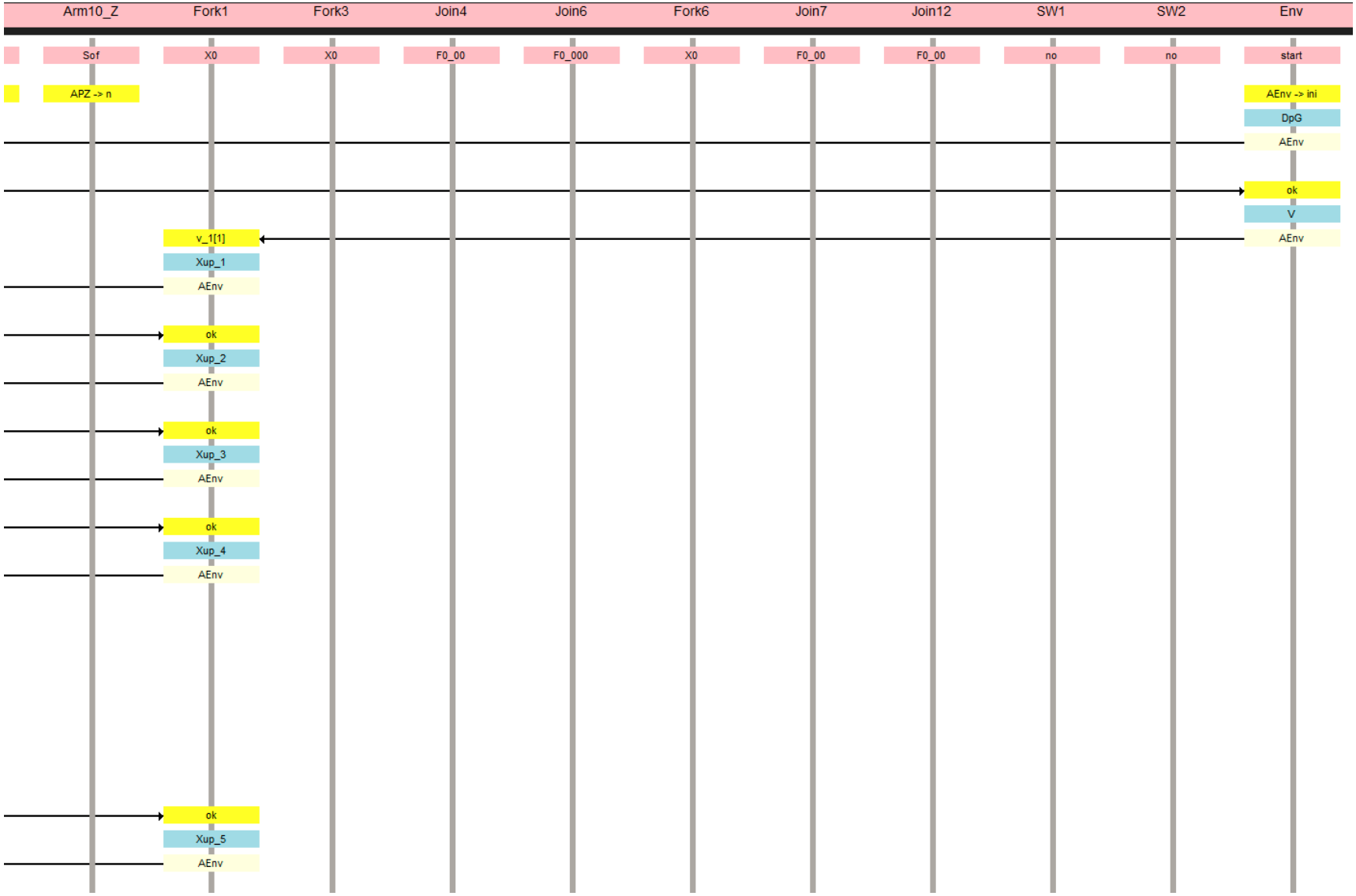
Potwierdza to prawidłowość zaprojektowania układu i powiązania przy założeniu braku usterek elementów i prawidłowej jeździe pociągów. Dla każdego z przeprowadzonych badań świadek bądź kontrprzykład w postaci graficznej stanowi dość obszerny rysunek. Wygląd świadka bez powiększenia, dla badania dla jazdy do stacji pociągu najdłuższego, przy braku usterek i w oparciu o blokadę (eksperyment 2) przedstawia rysunek 32, natomiast powiększenia dolnego i górnego lewego rogu (początek i koniec sekwencji) znajdują się na rysunkach 33 i 34. Wygląd kontrprzykładu bez powiększenia, dla eksperymentu 5 przedstawia rysunek 35, natomiast powiększenia dolnego i górnego lewego rogu (początek i koniec sekwencji) znajdują się na rysunkach 36 i 37.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21



Rysunek 32. Graficzna reprezentacja całego pliku tzw. świadka jednego z badań układu . Źródło: plik wygenerowany przez środowisko Dedan

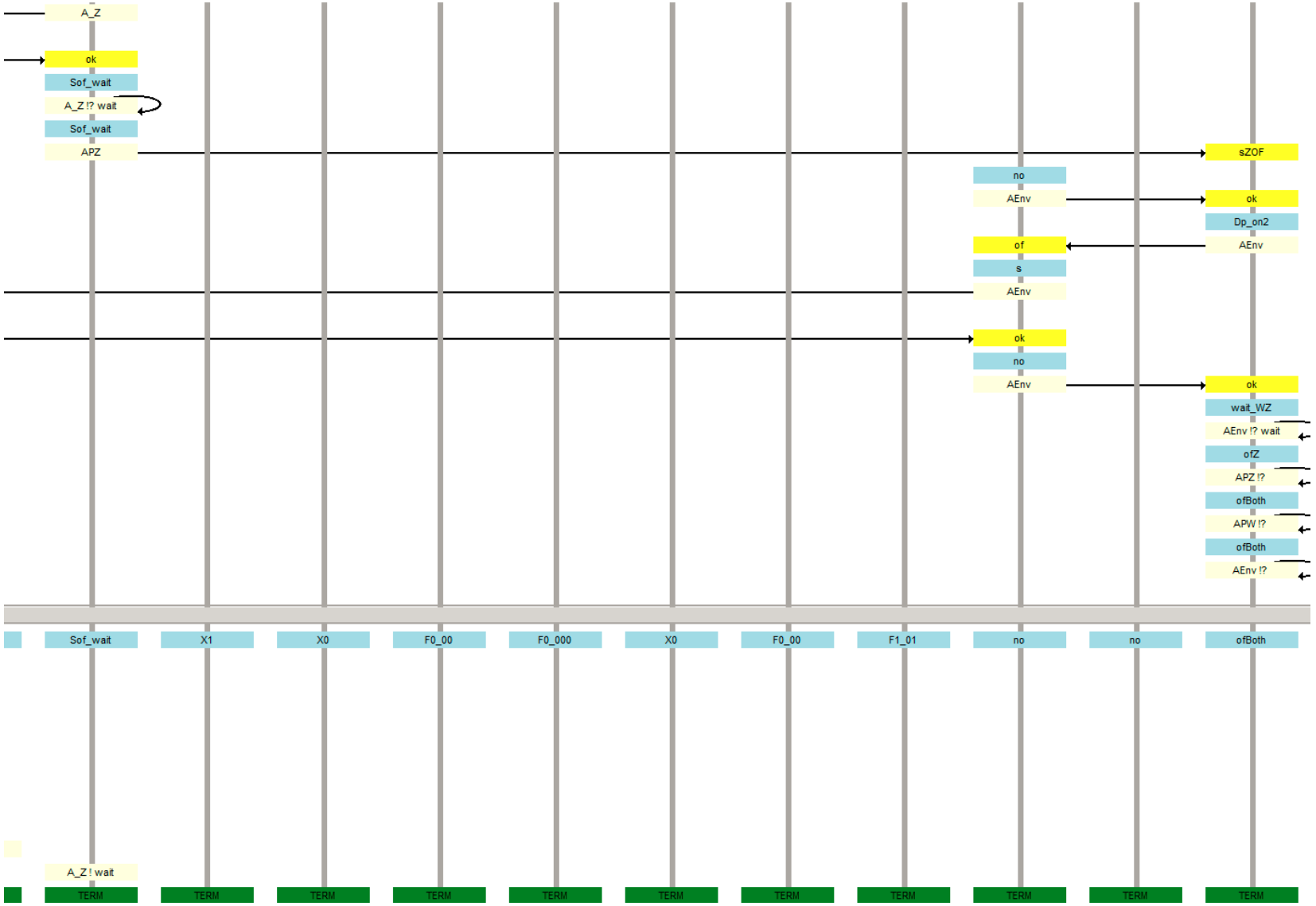
Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21



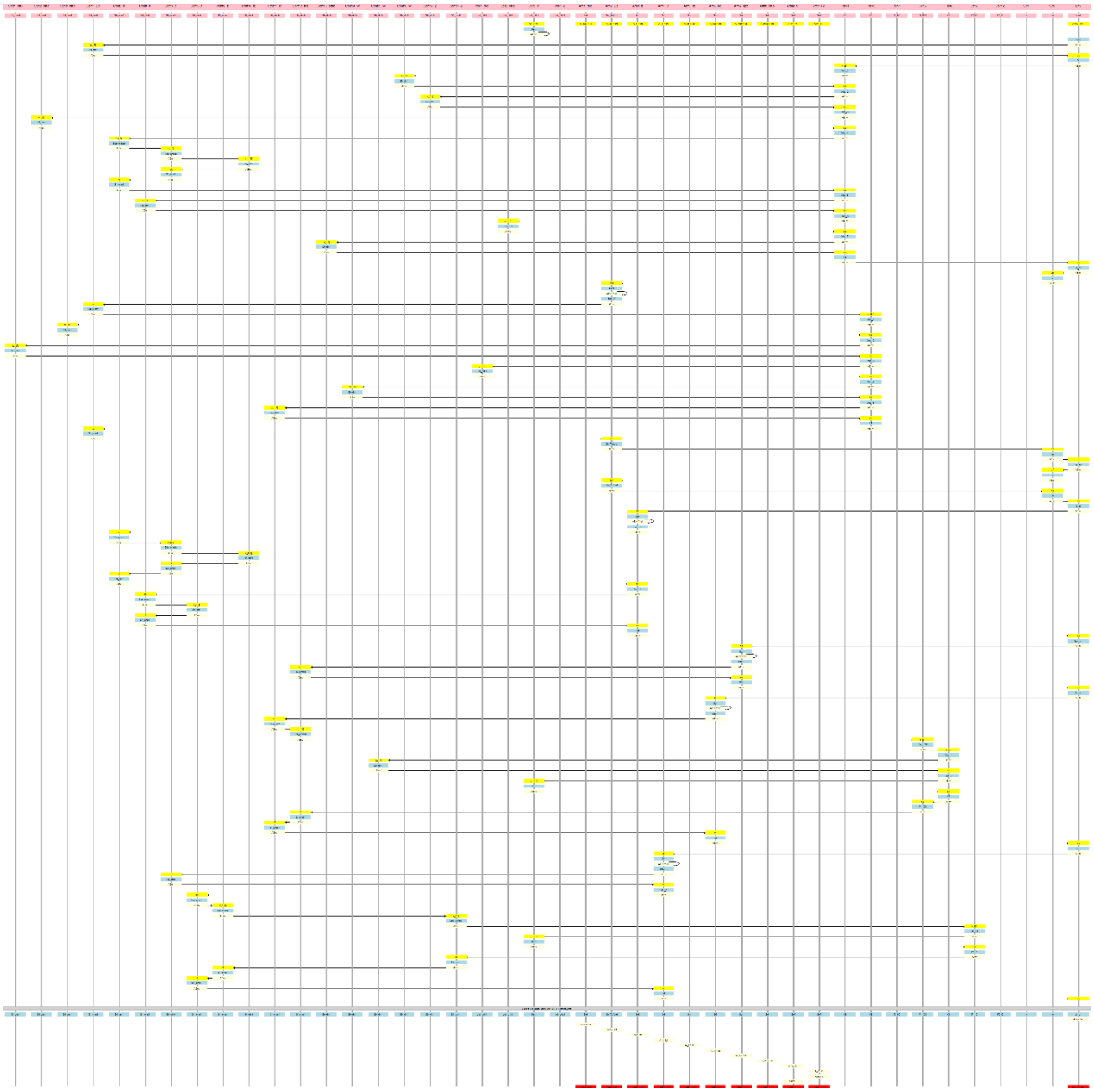
Rysunek 33. Fragment pliku tzw. świadka dla jednego z badań układu (górny prawy róg). Źródło: plik wygenerowany przez środowisko Dedan

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology



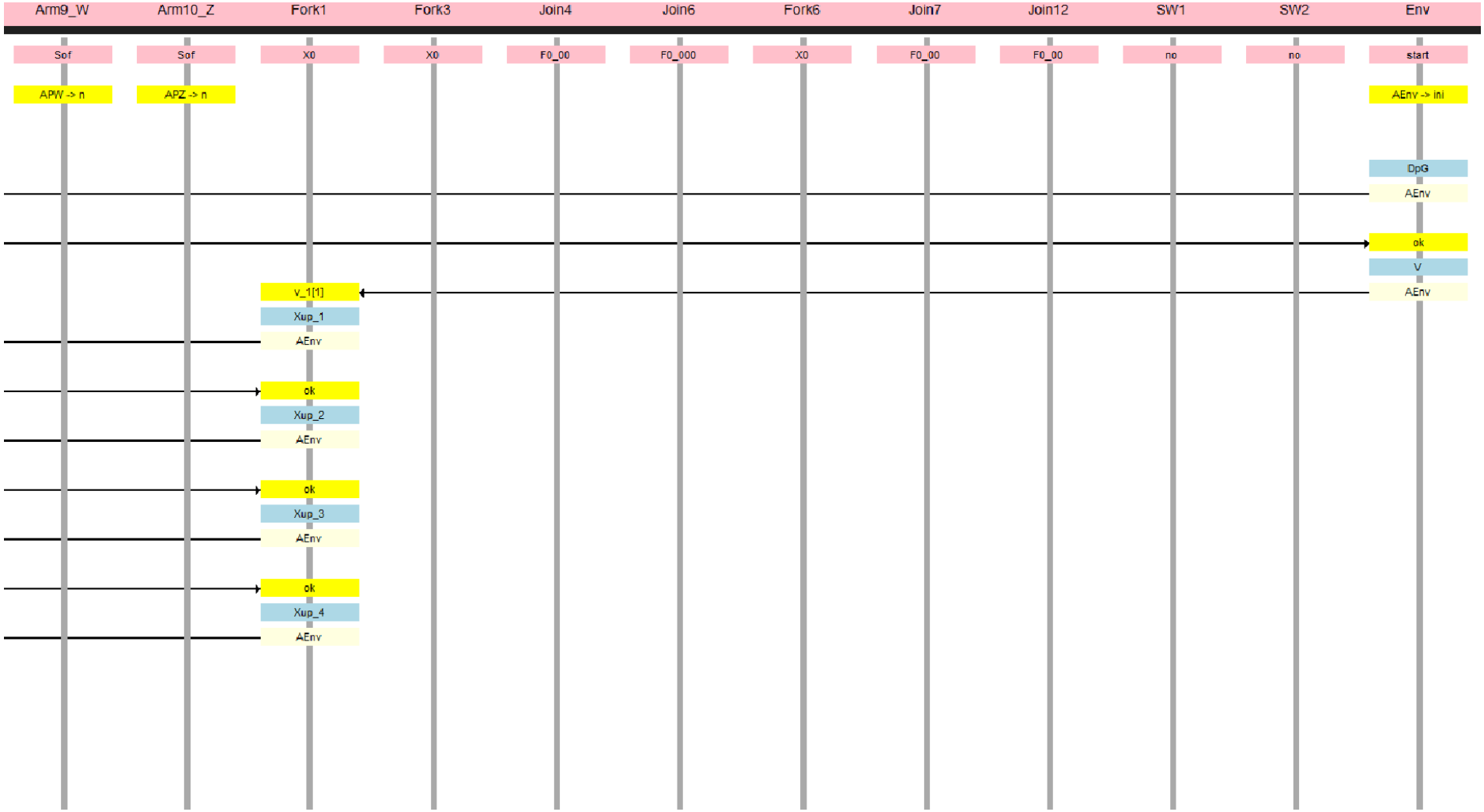
Rysunek 34. Fragment pliku tzw. świadka dla jednego z badań układu (dolny prawy róg). Źródło: plik wygenerowany przez środowisko Dedan



Rysunek 35. Graficzna reprezentacja całego pliku tzw. kontrprzykładu jednego z badań układu . Źródło: plik wygenerowany przez środowisko Dedan

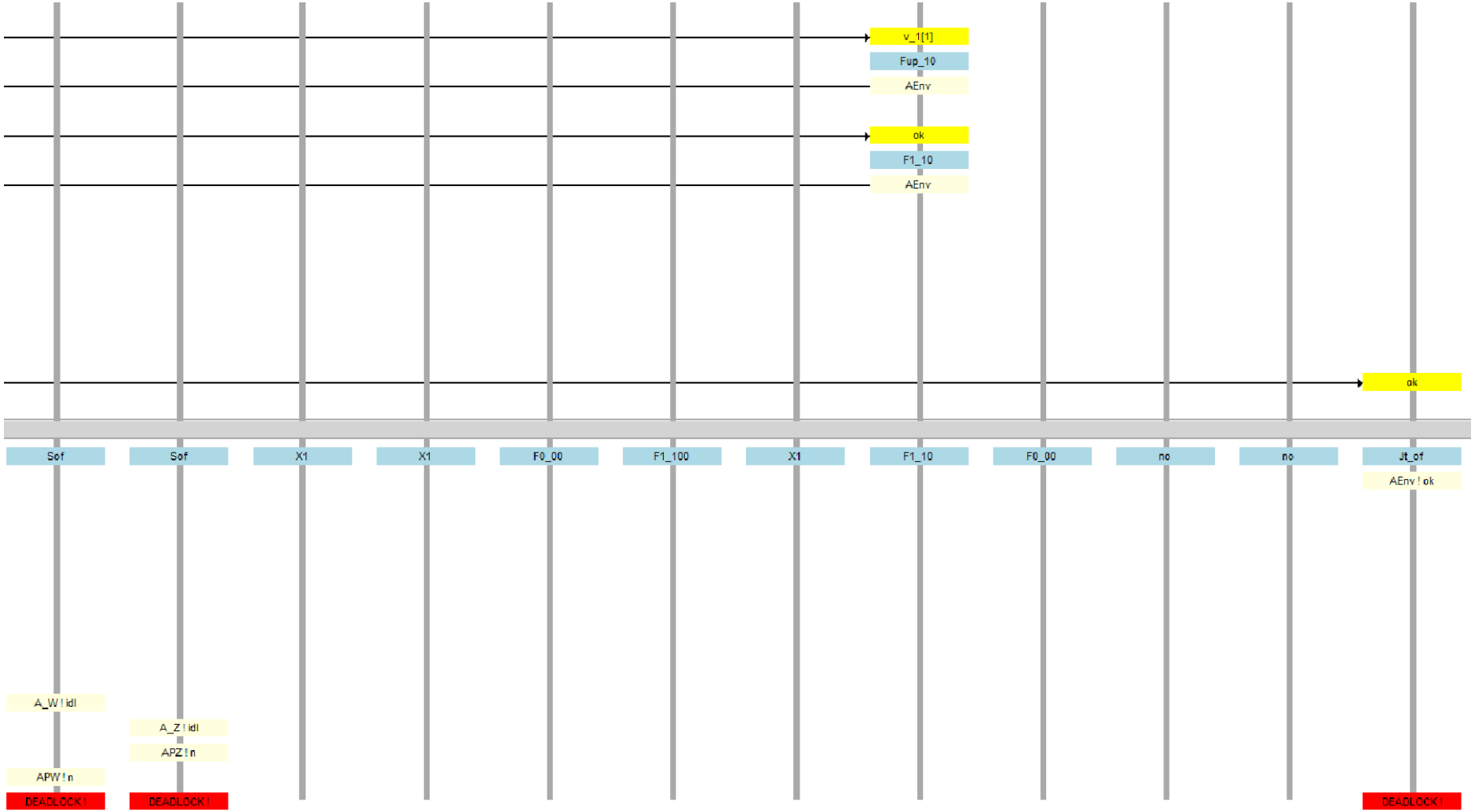
Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology



Rysunek 36. Fragment pliku tzw. kontrprzykładu dla jednego z badań układu (górny prawy róg). Źródło: plik wygenerowany przez środowisko Dedan

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology



Rysunek 37. Fragment pliku tzw. kontrprzykładu dla jednego z badań układu (dolny prawy róg). Źródło: plik wygenerowany przez środowisko Dedan

Zakleszczenie wystąpiło w eksperymentach 5 i 6 (eksperymenty z uszkodzeniami). Oznacza to nie osiągnięcie zdefiniowanego oczekiwanego, opisanego w modelu środowiska i obserwatorów działania. W przypadku eksperymentów 5 i 6 przekaźnik W albo nie wzbudził się wcale, albo odwzbudził przedwcześnie (przed wzbudzeniem przekaźnika Dp). Taki wynik eksperymentu potwierdza, że układ został zaprojekowany nieprawidłowo, czyli przy braku uwzględnienia skutków wprowadzonego uszkodzenia. Skutki takie powinny zostać uwzględnione.

W rzeczywistości skutkami przerwy cewki przekaźnika W jest brak możliwości jego wzbudzenia lub odwzbudzenie opisane dalej, w rozdziale 8.1.1, skutkujące pozostaniem blokady w stanie jak przed wjazdem pociągu na stację. Jeżeli funkcja ta, realizowana przez układ, który nie był przedmiotem modelu, zostałby uwzględniona w opisie środowiska modelu, wówczas eksperyment zakończyłby się terminacją agentów i brakiem zakleszczeń. Przykłady te obrazują możliwość identyfikacji błędów projektowania układów (brak uwzględnienia realizacji istotnej funkcji).

W przypadku eksperymentu 7 nie doszło do zakleszczenia. Eksperyment ten pokazuje usterkę, która została zmagazynowana i wymaga przeprowadzenia innego eksperymentu, w którym przekaźnik W będzie uszkodzony od początku.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

7 Omówienie rodzajów uszkodzeń elementów interfejsu

7.1 Wprowadzenie

W niniejszym rozdziale przedstawiono rodzaje podstawowych uszkodzeń elementów elektrycznych stosowanych w układach powiązań i dla których wykonano modele w rozdziale 4. Dla wybranych uszkodzeń z przytoczonych w rozdziale 7.2 zostały wykonane modele przedstawione w rozdziałach 4.6, 4.7.

W rozdziale 7.2 przedstawiono rodzaje uszkodzeń formalnie zdefiniowanych w normie PN-EN 50129 [53]. Ponieważ norma ta nie podaje praktycznego uzasadnienia dla wystąpienia określonego uszkodzenia, w rozdziale 7.3 omówiono i uzasadniono, kierując się własną praktyką, wybrane rodzaje uszkodzeń.

7.2 Źródło klasyfikacji i rodzaje uszkodzeń

W załączniku C normy 50129 [53] przedstawiono identyfikację wiarygodnych rodzajów uszkodzeń podzespołów sprzętowych. Podzespoły pogrupowano. Dla grupy zespołów połączeniowych jako prawdopodobne wymienia się w tablicy C-10 załącznika C normy 50129 [53]:

- Dla złącza:
    - Przerwę jednego zestyku lub większej liczby zestyków,
    - Przerwę ekranu,
    - Zwarcie lub zmniejszenie rezystancji izolacji między zestykami (*35, 36),
    - Zwarcie lub zmniejszenie rezystancji izolacji między zestykiem i obudową złącza (*35, 36),
    - Niewłaściwe usytuowanie mechaniczne złącza (*37),
- Dla kabli i przewodów:
    - Przerwę lub zwiększenie rezystancji jednego lub wielu przewodów,
    - Przerwę lub zwiększenie rezystancji ekranu (*38),
    - Zwarcie lub zmniejszenie rezystancji izolacji między dwoma przewodami lub większą ich liczbą (*39),
    - Zwarcie lub zmniejszenie rezystancji izolacji między przewodem lub przewodami i ekranem (*39),
    - Zwarcie lub zmniejszenie rezystancji izolacji między przewodem lub przewodami albo ekranem i zewnętrznymi częściami przewodzącymi (*39),

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

    - Wielokrotne przerwy i zwarcia (*39),
- Dla połączeń lutowanych, spawanych, owijanych, zaginanych, zaciskanych, skręcanych:
    - Przerwę,
    - Zwiększenie rezystancji.

Spełnienie poniższych warunków czyni pomijalnie małym prawdopodobieństwo usterek oznaczonych powyżej gwiazdką i numerem:

- Uwaga *35: „Materiał izolacyjny powinien być trwały. Odstępy izolacyjne powietrzne i odstępy izolacyjne powierzchniowe powinny co najmniej spełniać wymagania dotyczące izolacji wzmocnionej wg EN 50124-1."
- Uwaga *36: „Złącze powinno być solidnie skonstruowane. Wszystkie części złącza powinny być pewnie zamocowane."
- Uwaga *37: „Należy zapobiegać niepoprawnej orientacji złącza lub wetknięciu do niewłaściwego gniazda takimi środkami jak konstrukcja mechaniczna lub kodowanie układu końcówek. Alternatywnie, należy zapobiegać zagrożeniom wynikającym z niepoprawnego wetknięcia takimi środkami jak elektryczne kodowanie końcówek złącza lub przypisanie jednoznacznych adresów/identyfikatorów. Ryzyko powinno być ponadto ograniczone takimi środkami jak etykiety ostrzegawcze i szkolenie personelu."
- Uwaga *38: „Ekran powinien być skonstruowany solidnie i chroniony przed nadmiernym niszczeniem fizycznym. Połączenia ekranu powinny być solidne i trwale zamocowane".
- Uwaga *39: „Zapewniona powinna być dostatecznie solidna izolacja. Odstępy izolacyjne powietrzne i odstępy izolacyjne powierzchniowe powinny co najmniej spełniać wymagania dotyczące izolacji wzmocnionej wg EN 50124-1. Należy zapewnić ochronę przed nadmiernym niszczeniem fizycznym. Należy zapewnić ochronę przed obcymi ciałami przewodzącymi."

Dla łączników i przycisków w tablicy C-12 załącznika C normy 50129 [53] wymieniono następujące rodzaje uszkodzeń:

- Przerwa któregokolwiek zestyku,
- Zwarcie lub zmniejszenie rezystancji izolacji:
    - gdy zestyk jest otwarty (*20),
    - między zestykami (*20),

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

    - między zestykiem i obudową (*20),
- Sczepienie styków (*21),
- Zwiększenie rezystancji zestyku,
- Zacięcie się urządzenia w stanie przewodzenia,
- Drganie zestyku.

Dla przekaźników w tablicy C-3 załącznika C normy 50129 [53] wymieniono następujące rodzaje uszkodzeń:

- Przerwa którejkolwiek cewki,
- Przerwa w którymkolwiek zestyku,
- Zwarcie lub zmniejszenie rezystancji izolacji:
    - przy zestykach otwartych (*20),
    - między cewkami (*16),
    - między cewką i zestykiem (*20),
    - między cewką i obudową (*16),
    - między zestykami (*20),
    - między zestykiem i obudową (*20),
- Sczepienie styków (*21),
- Zwiększenie rezystancji zestyku,
- Drganie zestyku,
- Zwiększenie prądu przyciągania,
- Zmniejszenie prądu przyciągania (*22),
- Zwiększenie prądu zwalniania,
- Zmniejszenie prądu zwalniania (*22),
- Zmiana stosunku prądu przyciągania do prądu zwalniania (*22),
- Zwiększenie czasu przyciągania,
- Zmniejszenie czasu przyciągania (*22),
- Zwiększenie czasu zwalniania (*22),
- Zmniejszenie czasu zwalniania (*22),
- Przekaźnik nie przyciąga,
- Przekaźnik nie zwalnia (*22),
- Zamknięcie (przejściowe lub ciągłe) jakichkolwiek zestyków zwiernych w tym samym czasie, gdy zamknięte są jakiekolwiek zestyki rozwierne (*22),

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- Brak współbieżności zestyków zwiernych,
- Brak współbieżności zestyków rozwiernych.

Spełnienie poniższych warunków czyni pomijalnie małym prawdopodobieństwo usterek oznaczonych powyżej gwiazdką i numerem:

- Uwaga *16: „Odstępy izolacyjne powietrzne i odstępy izolacyjne powierzchniowe powinny co najmniej spełniać wymagania dotyczące izolacji wzmocnionej wg EN 50124-1. Wszystkie uzwojenia i połączenia powinny być pewnie umocowane. Moc tracona w elemencie powinna być wystarczająco ograniczona w celu ochrony przed wewnętrznym zwęglaniem się tego elementu (także w warunkach wystąpienia defektu).”
- Uwaga *20: „Wszystkie części mechanizmu przekaźnika lub przełącznika powinny być prawidłowo skonstruowane i pewnie zamocowane, przy czym uwzględnia się:
    - mechanizm działania,
    - system połączeń,
    - obwód magnetyczny (jeśli występuje),
    - cewkę (cewki) (jeśli występują).

  Odstępy izolacyjne powietrzne i odstępy izolacyjne powierzchniowe powinny spełniać co najmniej wymagania dotyczące izolacji wzmocnionej wg EN 50124-1.”
- Uwaga *21: „Materiały styków powinny być tak dobrane, by nie miały możliwości sczepiania się. Ryzyko sczepiania powinno być ponadto ograniczone przez odpowiedni projekt mechaniczny i konstrukcję zestyków. Prąd maksymalny powinien być ograniczony, w celu zapewnienia, że temperatura zestyków nie przekroczy wartości, przy której może nastąpić sczepienie.”
- Uwaga *22: „Stabilność charakterystyki przekaźnika powinna być zapewniona przez wnikliwe zwrócenie uwagi na następujące czynniki:
    - charakterystyka magnetyczna:
        - wybór materiału magnetycznego,
        - zapewnienie elementu zapobiegającego trwałemu namagnesowaniu obwodu magnetycznego (rdzenia),
        - ochrona przed zewnętrznymi polami magnetycznymi,
    - charakterystyka elektryczna:
        - wybór i jakość przewodu i izolacji,
        - jakość uzwojenia cewki,

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

        - jakość wyprowadzeń,
    - charakterystyka mechaniczna:
        - wybór i jakość materiałów,
        - pewność mocowania wszystkich części,
        - pewne zachowanie wszelkich regulacji związanych z bezpieczeństwem,
        - zapewnienie odpowiedniej siły powrotnej, z wykorzystaniem siły grawitacji (uzupełnianej, jeśli potrzebne, sprężynami i/lub sprężystością zestyków),
        - projekt i konstrukcja mechanizmu działania powinna być taka, by nie mógł się zablokować."

Podobne klasyfikacje uszkodzeń można odnaleźć w normie [54].

Należy zwrócić uwagę, że normy [53][54] wskazujące klasyfikacje uszkodzeń elementów nie określają:

- prawdopodobieństwa zaistnienia poszczególnych uszkodzeń,
- przyczyny poszczególnych uszkodzeń.

W zakresie prawdopodobieństwa zaistnienia uszkodzenia poszczególnych elementów interfejsu pewnym źródłem pozwalającym na określenie intensywności ich uszkodzeń jest podręcznik wojskowy [50]. Podane tam wartości intensywności uszkodzeń λ poszczególnych elementów modyfikowane są współczynnikami π, wartości których zależą m.in. od jakości wykonania i od warunków eksploatacji elementu, będącego przedmiotem obliczeń. Niestety podręcznik ten nie przewiduje określenia intensywności dla elementów składowych urządzenia takiego jak przekaźnik. Podane tam parametry dotyczą całego przekaźnika jako elementu mogącego się uszkodzić, bez wskazywania osobno intensywności uszkodzeń polegających np. na zgrzaniu zestyku lub przerwie w cewce.

Posiadanie dokładniejszych danych o intensywności uszkodzeń poszczególnych elementów dla określonego rodzaju uszkodzeń pozwalałoby przyjąć gradację prawdopodobieństwa rodzajów uszkodzeń, a znając skutki wynikające z działania modelowanego systemu określać, które uszkodzenia powinny być w ogóle rozważane, a które z rozważanych wymagałyby modyfikacji urządzenia w taki sposób, aby ograniczyć skutki takiego uszkodzenia.

Tymczasem w projektach powiązań urządzeń sterowania ruchem kolejowym przy braku omawianych danych przyjmuje się za wystarczające dwa działania:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- zerojedynkowe określenie czy uszkodzenie jest w ogóle prawdopodobne,
- zastosowanie kodeksów postępowania.

Określenie, czy uszkodzenie jest w ogóle prawdopodobne, to działanie polegające na sprawdzeniu cech elementów, z których skonstruowane jest urządzenie i warunków jego eksploatacji. Jeśli cechy i warunki te pozwalają spełnić określoną uwagę z uwag sformułowanych do rodzajów uszkodzeń, to uszkodzenie, dla którego potwierdzono uwagi można uznać za nieprawdopodobne i dalej nie zajmować się sprawdzaniem jego skutków.

Zastosowanie kodeksów postępowania polega natomiast na przyjęciu i zastosowaniu pewnych podstawowych zasad projektowania i instalacji ograniczających wpływ usterek na niepożądane działanie systemu. Przykładem takiego kodeksu może być rozdział 6.2 opracowania [78], gdzie jedna z przedstawionych tam reguł jest następująca:

„Należy unikać umieszczania zestyków przekaźnika klasy niższej niż C w układzie realizującym funkcje zależnościowe m.in. ze względu na możliwość jednoczesnego zwarcia zestyku zwiernego i rozwiernego takiego przekaźnika."

Reguła ta wynika z możliwości zastosowania w urządzeniach srk przekaźników, które nie spełniają wymaganej normami funkcji sztywnego prowadzenia zestyków.

7.3 Omówienie podstawowych rodzajów uszkodzeń

W zakresie przyczyn poszczególnych uszkodzeń można wskazać następujące.

Przerwy elementów przewodzących wynikać mogą z:

- mechanicznego oddziaływania na elementy – obluzowanie się połączenia wskutek udaru mechanicznego lub wystarczająco silnych lub długotrwałych drgań, przerwania obwodu przez człowieka: celowego (sabotaż, kradzież kabli) lub przypadkowego (podczas prac budowlanych), przerwania wskutek działania zwierzęcia (np. gryzonia), zużycia się elementów trących o siebie w trakcie montażu i demontażu,
- pokrycia się elementów łączących warstwą nieprzewodzącą (przy oddziaływaniu substancji chemicznych obecnych w atmosferze np. siarczków ze spalin, oparów wydzielanych z powłok[16]),

[16] Jedną z wad systemu JZH-111 wprowadzonego w Polsce w latach 70' było takie wykonanie wewnątrz zestawów izolacji przewodów doprowadzanych do transformatorów, że przy ich grzaniu wskutek przepływu prądu uwalniały się z nich opary syciwa, a następnie osiadały na zestykach przekaźników znajdujących się w tym samym zestawie powodując wzrost rezystancji lub brak ich przewodzenia.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

- przekroczenia wartości prądu płynącego przez element (przetopienie połączenia).

Przerwa przewodu, zestyku, połączenia powoduje brak zasilania elementów, do których zasilanie było doprowadzane uszkadzanym (przerywanym) elementem.

Zwiększenie rezystancji przy układach przełączających powoduje najczęściej niewystarczające zasilenie odbiornika energii elektrycznej (np. cewki przekaźnika, transoptora, lampki pulpitu) i wobec tego można je w określonych warunkach analizować tak, jak przerwę.

Zwarcie elementów może wynikać z różnych przyczyn. Można wskazać następujące:

- uszkodzenie izolacji wskutek oddziaływania mechanicznego (jak wyżej dla przerwy),
- środowisko pracy elementów pod napięciem: wilgoć połączona z brudem przewodzącym prąd (np. sproszkowanym węglem).

Zespawanie (zgrzanie) zestyku może powstać w wyniku przepływu przez zestyk prądu o dostatecznie dużym natężeniu. Skutki takiego zgrzania są następujące:

- Zestyk przewodzi (przekazuje sygnał pomiędzy wyprowadzeniami) pomimo próby oddziaływania na niego mechanicznie,
- Jeżeli zestyk był mechanicznie połączony z innymi zestykami i to połączenie realizowało funkcję sztywnego prowadzenia, wówczas uszkodzenie to powoduje wpływ na działanie pozostałych zestyków.

Drganie zestyku może wynikać z dwóch powodów:

- Oddziaływania na przekaźnik drgań lub udarów pochodzących z zewnątrz (z otoczenia). W tym zakresie przekaźniki powinny być odporne dzięki swojej konstrukcji lub dzięki właściwemu osadzeniu ich w miejscu ich instalacji (większym aparacie, stojaku itp.). Wartości drgań i udarów są sformułowane w normach [51][54][56][61] i instrukcjach wewnętrznych zarządcy infrastruktury [71].
- Normalnej pracy zestyku przekaźnika. Podczas zamykania zestyku następuje wielokrotne odbicie się styczek. Dopuszcza się taką pracę w ograniczonym interwale podczas zamykania zestyku (przykładowo do 50ms), co jest sformułowane jako wymaganie na konstrukcję przekaźnika [52][54][56].

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

8 Podsumowanie

8.1 Walidacja metody

Walidacja metody przebiegła w postaci dwóch etapów:

- Walidacji modeli ogólnych,
- Określenia jednakowości efektów analizy ręcznej obwodu z wynikami eksperymentu na przedstawionym modelu.

8.1.1 Ręczna analiza pracy obwodu

Działanie sprawnego obwodu będącego przedmiotem modelowania a następnie eksperymentów zostało opisane w rozdziale 6.6.3. Pliki świadków eksperymentów od 1 do 4 potwierdzają takie działanie.

Działanie obwodu z uszkodzeniem cewki przekaźnika W powinno być następujące.

Jeżeli do uszkodzenia cewki przekaźnika W dochodzi przed jego wzbudzeniem, to przekaźnik ten w toku pracy obwodu nie wzbudzi się. W konsekwencji nie wzbudzi się również przekaźnik Z. Niemożliwe będzie zwolnienie blokady, a więc wzbudzenie przekaźnika Dp.

Jeżeli do uszkodzenia cewki przekaźnika W dochodzi po jego wzbudzeniu ale przed odwzbudzeniem wywołanym wzbudzeniem przekaźnika Dp, to przekaźnik ten (W) wskutek tej usterki odwzbudza się. To z kolei wpływa na przekaźnik Z, który albo też się odwzbudza jeśli był już wzbudzony po wjeździe pociągu, albo uniemożliwia jego wzbudzenie. Odwzbudzony którykolwiek z przekaźników W i Z uniemożliwia wzbudzenie przekaźnika Dp, czyli zwolnienie blokady. Blokada pozostaje w stanie ustawionego kierunku.

Model z uszkodzeniem W nie wzbudza W ani Z, co prowadzi do zakleszczenia. Analiza kontrprzykładu umożliwia dotarcie do przyczyny zakleszczenia, a więc przyczyny nieprawidłowego działania układu.

8.1.2 Walidacja modeli ogólnych

Walidację modeli ogólnych przeprowadzono w toku eksperymentów modelu powiązania. Powstałe dokumenty świadków i kontrprzykładów (dla 7 eksperymentów) pozwoliły na określenie prawidłowości zachowania poszczególnych elementów modelu zarówno w przypadku modeli sprawnych, jak i uszkodzonych.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

8.2 Wnioski i kierunki dalszych prac

W przedstawionej w niniejszej pracy metodologii udało się skutecznie zamodelować przekaźnikowe układy srk i ich powiązania z urzadzeniami komputerowymi, wykorzystując znany formalizm IMDS oraz istniejące wcześniej i dostępne środowisko Dedan. Układy przekaźnikowe posiadają wiele zagadnień sprawiających trudności w ich weryfikacji. Przykładowo:

- zawierają wiele elementów analogowych, przy pomocy których projektanci w niejednorodny sposób osiągają określone własności;
- ich elementy realizują skomplikowane funkcje logiczne, a logika układu nie jest czysto dwuwartościowa (przekazanie i wycofanie masy i zasilania stanowi w gruncie rzeczy cztery wartości logiczne, a do tego jeszcze należy uwzględnić logikę przełączeń elementów sterowanych manualnie);
- skuteczne, w gruncie rzeczy kombinacyjne zachowanie elementów jest realizowane przy pomocy elementów pamięciowych, jak styki sterowane z cewki tego samego przekaźnika;
- układy zawierają elementy, które mogą zadziałać „dwukierunkowo", to znaczy przekazywać napięcie zarówno w jednym kierunku jak i w drugim; zasadniczo rzadko kiedy może to się stać w tej samej chwili, ale należałoby ten fakt udowodnić;
- analizowane części układów zawierają „fragmenty" elementów, na przykład styki przekaźnika, którego cewka jest w innej części, lub odwrotnie.

Trudności wskazał przegląd literatury, w której weryfikowane układy składają się najwyżej z kilku elementów (najczęściej: dwa przekaźniki o niewielkiej liczbie styków). Opisane powyżej trudności zidentyfikowano również w toku prac i odpowiednio potraktowano dla osiągnięcia celu modelowania.

Zastosowanie parametryzowanych prototypów elementów i preprocesora, który tłumaczy prototypy na modele, oraz dekompozycja przekaźników na trzy podelementy pozwoliło na zmniejszenie liczby elementów składowych modelowanego ukladu elektycznego do pięciu: cewka, kotwica, przełącznik/zestyk, połączenie galwaniczne doprowadzeń i rozgałęzienie doprowadzeń. To z kolei pozwoliło na zwiększenie złożoności weryfikowanego układu (ujęcie większej liczby przekaźników i zestyków w modelowanym układzie). W przeanalizowanej literaturze, w przeciwieństwie do zastosowanej w rozprawie metodzie, modele są budowane manualnie na podstawie obserwacji schematu. Istotne jest również wykonane w niniejszej pracy

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

zaprojektowanie prototypów elementów w postaci graficznej automatów DA^3, w pełni zgodnej z semantyką modelu matematycznego IMDS.

Dodatkowym aspektem poruszonym w niniejszej pracy jest sposób zadawania pytań o działanie układów. W literaturze większość podejść opiera się o specyfikowanie własności w logice temporalnej (zadawanie pytań w postaci zdań logicznych), co jest w gruncie rzeczy niedostępne dla inżyniera nie posiadającego odpowiedniej wiedzy matematycznej. W niewielu przypadkach stosuje się, tak jak w niniejszej pracy, automaty modelujące wymagane zachowanie.

W żadnym z przeanalizowanych źródeł autorzy nie badają bezpieczeństwa układu w razie jego awarii (w tym awarii dynamicznej w czasie działania układu) lub niepoprawnej sekwencji sygnałów wejściowych (np. jazda pociąg wstecz). Własności te udało się wyrazić w niniejszej pracy jako automaty obserwujące w połączeniu z badaniem ich cech takich jak zakleszczenie, nieuchronna terminacja i możliwa terminacja. Opracowano również modele elementów uszkodzonych. Na podkreślenie zasługuje fakt, że cechy te są w środowisku Dedan weryfikowane automatycznie nawet dla pojedynczych elementów układu lub niewielkich podzespołów (częściowe zakleszczenia i częściowa terminacja). Jest to unikatową cechą weryfikatora tak, jak sprawiedliwość weryfikacji zabezpieczająca przed wykrywaniem fałszywych zakleszczeń.

W przeanalizowanej literaturze udało się znaleźć dwa rozwiązania stosujące automaty obserwujące, podobnie jak w niniejszej pracy, ale nadal wymagają one do weryfikacji zastosowania formuł temporalnych, tyle że nie do samego układu a do obserwatora. Zastosowane w pracy środowisko Dedan nie wymaga natomiast od użytkownika konstruowania formuł temporalnych.

Wykorzystanie opisanych modeli i narzędzi do weryfikacji przekaźnikowych układów srk i ich powiązania z urzadzeniami komputerowymi jest podejściem oryginalnym. Zostało ono po raz pierwszy omówione w artykule [35] współautorstwa autora niniejszej dysertacji a w niniejszej rozprawie rozwinięte.

Wykonując działania opisane w rozdziale 6.6, na podstawie działań opisanych ogólnie w rozdziale 1.10 i z wykorzystaniem opracowanych modeli zawartych w rozdziale 4 udało się potwierdzić tezę. Cel pracy został osiągnięty.

Dalsze możliwości i potrzeby rozwoju zaproponowanej metody można przedstawić w następujący sposób:

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Potrzeby rozwoju narzędzia:

- Dodanie kolejnych modeli elementów, np. oporników, kondensatorów, diod prostowniczych, transformatorów, lampek (elementów sygnalizacji optycznej).
- Opracowanie modeli dwukierunkowych do przetwarzania sygnałów.
- Dodanie czasu przetwarzania konkretnych czynności np. przełączenia zestyków w przekaźniku.
- Opracowanie przeglądarki pliku xml kontrprzykładu lub świadka, pozwalającej wskazywać w danym wierszu kontrprzykładu lub świadka jaki jest stan poszczególnych elementów modelu.
- Opracowanie przeglądarki pliku xml kontrprzykładu lub świadka, pozwalającej wskazywać na rysunku obwodu stan elementów modelu, położenie agentów i komunikaty.

Świadek lub kotrprzykład jest generowany w postaci graficznej i pliku xml. Dane zawarte w pliku xml mogą być danymi wejściowymi do narzędzia analitycznego.

Można postulować sprawdzanie metodą przedstawioną w rozprawie również poprawności sformułowania przepisów.

Korzystnym byłoby opracowanie algorytmów i narzędzi pozwalających automatyzować pracę obserwatorów. Do tego celu należałoby poszukiwać prostego sposobu wprowadzenia formuł określających prawidłowość działania, aby automatycznie wykrywać czy modelowane urządzenie działa prawidłowo, czy nie. Do tego celu potrzeba również modeli operatorów urządzeń, pojazdów szynowych i ich prowadzących. Ogólnie – należałoby rozwijać modele otoczenia. Ich działanie można opisać wymaganiami formalnymi – przepisami, instrukcjami.

Należałoby również określić czasochłonność zastosowanej w niniejszej pracy metody i odnieść ją do czasochłonności innych metod i narzędzi umożliwiających badania przekaźnikowych układów srk i ich powiązania z urządzeniami komputerowymi.

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

9 Spis tabel

Tabela 1. Przykładowe przyporządkowanie określonym wartościom sygnałów elektrycznych sygnałów binarnych dla interfejsu S kodera LEU. Źródło: [36] 23
Tabela 2. Przykładowa tabela prawdy dla wskazań sygnalizatora zawierająca w komórkach stany komór wen. Źródło: opracowane własne na podstawie [36] 25
Tabela 3. Formuły czasowe do znajdowania różnych sytuacji w procesach. Źródło: [17]. 50
Tabela 4. Sygnały interfejsu. UP – układ pośredniczący, UK – urządzenia komputerowe. Źródło: opracowanie własne 100
Tabela 5. Relacje pomiędzy sygnałami. Źródło: opracowanie własne 102
Tabela 6. Przyporządkowanie określonym wartościom sygnałów elektrycznych sygnałów binarnych. Wejścia systemu komputerowego. Źródło – opracowanie własne na podstawie [33] 109
Tabela 7. Tabela prawdy - przyporządkowanie stanów blokady do kombinacji stanów wejść wektora **Y2** urządzeń komputerowych na stacji A. Źródło: opracowanie własne na podstawie [38] 111
Tabela 8. Przyporządkowanie określonym wartościom oporu elektrycznego zestyków sygnałów binarnych. Wyjścia systemu komputerowego. Źródło – opracowanie własne na podstawie [51] 120
Tabela 9. Lista aparatów badanego układu. Źródło: opracowanie własne na podstawie pliku wygenerowanego przez program SEE Electrical Expert 128
Tabela 10. Zestawienie połączeń dla badanego układu. Źródło: opracowanie własne na podstawie pliku wygenerowanego przez program SEE Electrical Expert 129
Tabela 11. Prawidłowa kolejność zajmowania i zwalniania odcinków podczas jazdy pociągu. Źródło: opracowanie własne 133

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

10 Spis rysunków

Rysunek 1. Idea współpracy urządzeń srk. 1-6 - możliwe interakcje. Źródło: [40] 14
Rysunek 2. Powiązania poziome (pomarańczowe) i pionowe (zielone) urządzeń ksrk w warstwowym modelu ksrk. Źródło: opracowanie własne na podstawie [77] 16
Rysunek 3. Warstwy powiązania i wykorzystywane do ich określenia zbiory dokumentów. Źródło: opracowanie własne. 20
Rysunek 4. Przykładowy graf reprezentujący możliwe zmiany stanów sygnałów odczytywanych poprzez koder LEU. Źródło [36] 26
Rysunek 5. Graficzne przedstawienie metody. Źródło: opracowanie własne. 33
Rysunek 6. Uszczegółowiony graf weryfikacji. Źródło: opracowanie własne 34
Rysunek 7. Pełny opis akcji jako łuk w grafie automatu, oraz jego uproszczenia. a – pełna akcja, b – uproszczona etykieta, c – etykieta bez serwera docelowego. Źródło: opracowanie własne na podstawie [35] 52
Rysunek 8. Wykres charakterystyki prądowej przekaźnika uwzględniający zjawisko histerezy. Źródło: opracowanie własne 58
Rysunek 9. Charakterystyka czasowa przełączania zestyków zwiernych przekaźnika. Źródło: opracowanie własne 60
Rysunek 10. Uogólniony model interfejsu przekaźnikowego w urządzeniach komputerowych w zakresie wyjść. Część 1 z 2. Źródło: opracowanie własne 64
Rysunek 11. Uogólniony model interfejsu przekaźnikowego w urządzeniach komputerowych w zakresie wyjść. Część 2 z 2. Źródło: opracowanie własne 65
Rysunek 12. Schemat poglądowy uzwojenia przekaźnika (po lewej). Sytuacja zastosowania wielu źródeł sygnałów sterujących (po prawej). Źródło: opracowanie własne na podstawie [35] 71
Rysunek 13. Graf automatu modelu uzwojenia przekaźnika. Źródło: opracowanie własne na podstawie [35] 73
Rysunek 14. Schemat poglądowy kotwicy przekaźnika: j – numer agenta kotwicy, hO – liczba zestyków NO, hC – liczba zestyków NC. Źródło: opracowanie własne na podstawie [35] 73
Rysunek 15. Graf automatu modelu kotwicy przekaźnika. j – numer agenta kotwicy, Y – styki NC, X – styki NO. Źródło: opracowanie własne na podstawie [35] 75
Rysunek 16. Model zestyku przekaźnika, przełącznika. Źródło: opracowanie własne na podstawie [35] 76
Rysunek 17. Graf automatu zestyku. Źródło: opracowanie własne na podstawie [35] 77
Rysunek 18. Graf automatu modelu uzwojenia z wystąpieniem przerwy. Źródło: opracowanie własne na podstawie [35] 79
Rysunek 19. Graf automatu modelu kotwicy przekaźnika ze spieczonym zestykiem zwiernym (numer 2). j – numer agenta kotwicy, Y – styki NC, X – styki NO. Źródło: opracowanie własne na podstawie [35] 80
Rysunek 20. Schematyczny układ torów i związana z nim lokalizacja urządzeń i powiązania. K, L, M, Z – semafory stacji A istotne dla powiązania, Jz11, JtZ, itAB, JtA – odcinki kontroli niezajętości istotne dla powiązania. Źródło: opracowanie własne na podstawie [35] 94
Rysunek 21. Ogólny szkic współpracujących systemów: komputerowego i blokady Eap. Źródło: opracowanie własne na podstawie [40] 95
Rysunek 22. Graf pracy blokady Eap z punktu widzenia obsługi poleceń z pulpitu i sygnałów wejściowych z otoczenia. Źródło: opracowanie własne na podstawie [34][38] 103
Rysunek 23. Harmonogram pracy przekaźników w blokadzie Eap podczas ustawiania kierunku ze stacji B do stacji A. Źródło: opracowanie własne na podstawie [38][8] 106
Rysunek 24. Legenda harmonogramu pracy przekaźników przedstawionego na rysunku 23. Źródło: opracowanie własne na podstawie [8] 107

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Rysunek 25. Graf stanów komputerowych urządzeń stacyjnych dla przypadku współpracy z blokadą Eap. Źródło: opracowanie własne na podstawie [38] i obserwacji pulpitów stanowiska laboratoryjnego 119
Rysunek 26. Graf dla sygnałów Kc i Kpz semafora wjazdowego. Źródło: opracowanie własne, na podstawie [2] 123
Rysunek 27. Graf dla sygnału S3_/Sz. Źródło: opracowanie własne 125
Rysunek 28. Graf dla sygnału S4_Jt. Źródło: opracowanie własne 126
Rysunek 29. Graf dla sygnału S6_/Uwy. Źródło: opracowanie własne 126
Rysunek 30. Schemat połączeń obwodu oddziaływania pociągów na blokadę. Źródło: opracowanie własne na podstawie [35][38] 127
Rysunek 31. Rozmieszczenie odcinków kontroli niezajętości it, Jt i Jz przy semaforze wjazdowym stacji. Źródło: opracowanie własne 133
Rysunek 32. Graficzna reprezentacja całego pliku tzw. świadka jednego z badań układu . Źródło: plik wygenerowany przez środowisko Dedan 138
Rysunek 33. Fragment pliku tzw. świadka dla jednego z badań układu (górny prawy róg). Źródło: plik wygenerowany przez środowisko Dedan 139
Rysunek 34. Fragment pliku tzw. świadka dla jednego z badań układu (dolny prawy róg). Źródło: plik wygenerowany przez środowisko Dedan 140
Rysunek 35. Graficzna reprezentacja całego pliku tzw. kontrprzykładu jednego z badań układu . Źródło: plik wygenerowany przez środowisko Dedan 141
Rysunek 36. Fragment pliku tzw. kontrprzykładu dla jednego z badań układu (górny prawy róg). Źródło: plik wygenerowany przez środowisko Dedan 142
Rysunek 37. Fragment pliku tzw. kontrprzykładu dla jednego z badań układu (dolny prawy róg). Źródło: plik wygenerowany przez środowisko Dedan 143

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

11 Literatura

[1] Aktualizacja dokumentu "Wymagania bezpieczeństwa dla systemów sterowania ruchem kolejowym" LA/11/10. Instytut Kolejnictwa, Warszawa 2010

[2] Album schematów przekaźnikowych urządzeń zabezpieczenia ruchu kolejowego typu E. Aktualizacja 1989. Centralne Biuro Projektowo – Badawcze Budownictwa Kolejowego. Warszawa 1989

[3] Amendola A. Becchi A., Cavada R. et al.: NORMA: a tool for the analysis of Relay-based Railway Interlocking Systems. In: TACAS 2022: Tools and Algorithms for the Construction and Analysis of Systems, Munich, Germany, 2–7 Apr 2022, 2022, pp. 125–142

[4] Assbury K., Góras S., Kempny S.: Międzystacyjna przewodowa blokada liniowa z wykorzystaniem podsystemów i elementów sbl-typu Eac, W: Automatyka kolejowa nr 3/1988

[5] Baier C., Katoen J.-P.: Principles of Model Checking. MIT Press: Cambridge, MA, USA, 2008. ISBN 9780262026499

[6] Bartczak M.: Szybki tyrystorowy układ nastawiania zwrotnicy na górce rozrządowej. W: Automatyka kolejowa, nr 6/1988

[7] Behrmann G., David A., Larsen K. et al.: Developing UPPAAL over 15 years. In: Software Practice and Experience, vol. 41, no. 2, pp. 133–142, Feb. 2011, doi:10.1002/spe.1006

[8] Buras K.: Analiza skutków i uszkodzeń blokady Eap. Praca dyplomowa inżynierska, promotor: dr inż. Przemysław Ilczuk, Politechnika Warszawska, Wydział Transportu, Warszawa 2022

[9] Cimatti A., Clarke E., Giunchiglia E.: et al.: NuSMV 2: An OpenSource Tool for Symbolic Model Checking. In CAV 2002: Computer Aided Verification, Copenhagen, Denmark, 27–31 July 2002; Brinksma, E., Larsen, K.G., Eds.; Springer: Berlin/Heidelberg, Germany, 2002. Volume 2404, pp. 359–364

[10] Clarke E., Grumberg O., Peled D.: Model Checking. MIT Press: Cambridge, MA, USA, 1999. ISBN 0-262-03270-8

[11] Daskaya I., Huhn M., Milius S.: Formal Safety Analysis in Industrial Practice. In: FMICS 2011: Formal Methods for Industrial Critical Systems, rento, Italy, 29-30 Aug 2011, 2011, pp. 68–84

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

[12] Daszczuk W., Bielecki M., Michalski J.: Rybu: Imperative-style Preprocessor for Verification of Distributed Systems in the Dedan Environment. In: Proceedings of the KKIO'17—Software Engineering Conference, Rzeszów, Poland, 14–16 September 2017.

[13] Daszczuk W.: Communication and Resource Deadlock Analysis using IMDS Formalism and Model Checking. The Computer Journal 2017, vol. 60, pp. 729–750, doi:10.1093/comjnl/bxw099

[14] Daszczuk W.: Evaluation of temporal formulas based on "Checking By Spheres." In: Proceedings of the Euromicro Symposium on Digital Systems Design, Warsaw, Poland, 4–6 September 2001. pp. 158–164, doi:10.1109/DSD.2001.952267

[15] Daszczuk W.: Fairness in Temporal Verification of Distributed Systems. In: 13th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, 2-6 July 2018, Brunów, Poland, AISC vol.761, 2019, pp. 135–150

[16] Daszczuk W.: Graphic modeling in Distributed Autonomous and Asynchronous Automata (DA3). In: Software and Systems Modeling, vol. 20, no. 5, pp. 363–398, 2021

[17] Daszczuk W.: Specification and Verification in Integrated Model of Distributed Systems (IMDS). Computers 2018, 7, 65. https://doi.org/10.3390/computers7040065

[18] Daszczuk, W.: Threefold Analysis of Distributed Systems: IMDS, Petri Net and Distributed Automata DA3. In: Proceedings of the 37th IEEE Software Engineering Workshop, Federated Conference on Computer Science and Information Systems (FEDCSIS'17), Prague, Czech Republic, 3–6 September 2017; pp. 377–386, doi:10.15439/2017F32.

[19] Dąbrowa-Bajon M.: Podstawy sterowania ruchem kolejowym. Funkcje, wymagania, zarys techniki. OWPW, Warszawa 2015

[20] de Almeida Pereira D., Deharbe D., Perin M. et al.: B-Specification of Relay-Based Railway Interlocking Systems Based on the Propositional Logic of the System State Evolution. In: RSSRail 2019: International Conference on Reliability, Safety, and Security of Railway Systems, Lille, France, 4–6 June 2019, 2019, pp. 242–258

[21] Dedan. Available online: http://staff.ii.pw.edu.pl/dedan/files/DedAn.zip (dostęp 24.10.2018)

[22] Dokumenty Baseline 3 release 1 udostępnione przez EULYNX on-line https://eulynx.eu/index.php/documents/published-documents Dostęp 26.11.2019

[23] Dyduch J., Kornaszewski M.: Systemy sterowania ruchem kolejowym. Wydawnictwo Politechniki Radomskiej, Radom 2007

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

[24] Fantechi A.: Distributing the Challenge of Model Checking Interlocking Control Tables. In: ISoLA 2012: International Symposium On Leveraging Applications of Formal Methods, Verification and Validation, Heraklion, Greece, 15-18 Oct 2012, 2012, pp. 276–289

[25] Ferrari, A., Magnani G., Grasso D. et al.: Model Checking Interlocking Control Tables. In: FORMS/FORMAT 2010, 8th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems, Braunschweig, Germany, 2-3 Dec 2010, 2011, pp. 107–115

[26] Haxthausen A., Le Bliguet M., Kjær A.: Modelling and Verification of Relay Interlocking Systems. In: Monterey Workshop 2008: Foundations of Computer Software, Budapest, Hungary, 24-26 Sept 2008, 2010, pp. 141–153

[27] Haxthausen A., Østergaard P.: On the Use of Static Checking in the Verification of Interlocking Systems. In: ISoLA 2016: International Symposium on Leveraging Applications of Formal Methods, Corfu, Greece, 10-14 Oct 2016, 2016, pp. 266–278

[28] Haxthausen A., Peleska J., Pinger R.: Applied Bounded Model Checking for Interlocking System Designs. In: SEFM 2013 Collocated Workshops on Software Engineering and Formal Methods, LNCS 8368, Madrid, Spain, 23-24 Sept 2013, 2014, pp. 205–220

[29] Holzmann G.: Tutorial: Proving properties of concurrent systems with SPIN. In: Proceedings of the 6th International Conference on Concurrency Theory (CONCUR'95), Philadelphia, PA, USA, 21–24 August 1995. pp. 453–455, doi:10.1007/3-540-60218-6_34

[30] Instrukcja konserwacji, przeglądów oraz napraw bieżących urządzeń sterowania ruchem kolejowym Ie-12 (E-24), PKP PLK S.A., Warszawa 2017

[31] Instrukcja obsługi przekaźnikowych urządzeń sterowania ruchem kolejowym Ie-10 (E-18). PKP PLK S.A., Warszawa 2005

[32] Instrukcja sygnalizacji Ie-1 (E-1), PKP PLK S.A., Warszawa 2019

[33] Jednoodstępowa (półsamoczynna) blokada liniowa typu Eap-94. Dokumentacja techniczno - ruchowa, P.P.H.U. Maciej Grot Sp. z o.o., Sosnowiec 2016

[34] Jednoodstępowa (półsamoczynna) blokada liniowa typu Eap-94. Instrukcja obsługi. P.P.H.U. Maciej Grot Sp. z o.o., Sosnowiec 2016

[35] Karolak J., Daszczuk W., Grabski W., Kochan A.: Temporal Verification of Relay-Based Railway Traffic Control Systems Using the Integrated Model of Distributed Systems. Energies 2022, 15, 9041.. https://doi.org/10.3390/en15239041

[36] Karolak J.: Interface and connection model in the railway traffic control system. Archives of Transport 2021, 58, s. 137–147. https://doi.org/10.5604/01.3001.0014.9086

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

[37] Kawalec P., Koliński D.: Zastosowanie języka VHDL do badania złożonych sieci zestykowych. W: Pomiary Automatyka Kontrola, vol. 54, nr 8, 2008 s. 529-531

[38] Kempny J.: Półsamoczynna blokada liniowa typu Eap-94. Album schematów. Sterowanie pulpit klasyczny. P.P.H.U. Maciej Grot Sp. z o.o., Sosnowiec 2016

[39] Klimov A., Kupriyanovsky V., Stepanenko A. et al.: BIM and engineering formalized ontologies on the European digital railway in the EULYNX-data economy. International Journal of Open Information Technologies, 2018, 6(8), pp. 38-65

[40] Kochan A., Konopiński L., Ilczuk P., Karolak J.: Wymagania formalno-prawne dotyczące badania interfejsów w systemach sterowania ruchem kolejowym. W: Problemy Kolejnictwa, 2015, 59 (168), s. 21-26

[41] Koczor J.: Program wspomagający projektowanie SEE Electrical Expert. W: Elektro systemy, nr 12/2005

[42] Koczor J.: Program wspomagający projektowanie. W: Elektroinstalator, nr 11/2004

[43] Kosiński A. i in.: Album schematów półsamoczynnej blokady liniowej typu Eap. Centralne Biuro Projektowo-Badawcze Budownictwa Kolejowego, Warszawa 1991/1994

[44] Lahtine J.: Model Checking Timed Safety Instrumented Systems. Helsinki, 2008

[45] Linh Ngoc Bui: An Analysis of the Benefits of EULYNX-style Requirements Modeling for ProRail. PhD Thesis. Eindhoven University of Technology. Eindhoven 2017

[46] Manna Z., Pnueli A.: The Temporal Logic of Reactive and Concurrent Systems. Springer, New York 1992

[47] Masticola S., Ryder B.: Static Infinite Wait Anomaly Detection in Polynomial Time. In: Proceedings of the 1990 International Conference on Parallel Processing, Urbana-Champaign, IL, USA, 13–17 August 1990; pp. 78–87

[48] Mickiewicz T., Zubkow A.: Przekaźnikowe urządzenia nastawcze typu półblokowego. WKiŁ, Warszawa 1961

[49] Mickiewicz T.: Elektryczne urządzenia zabezpieczenia ruchu pociągów. Urządzenia stacyjne. Wydawnictwa Komunikacyjne, Warszawa 1959

[50] MIL HDBK 217F Military Handbook, Reliability Prediction of Electronic Equipment, Department of Defense, Washington DC 20301, 02.12.1991

[51] Norma branżowa BN-76 3506-11 Urządzenia zabezpieczenia ruchu kolejowego. Przekaźniki. Ogólne wymagania i badania

[52] Norma EN 50205:2002 Relays with forcibly guided (mechanically linked) contacts

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

[53] Norma EN-50129: 2007 Zastosowania kolejowe. Systemy łączności, przetwarzania danych i sterowania ruchem, Elektroniczne systemy sterowania ruchem związane z bezpieczeństwem

[54] Norma IEC 62425:2007 Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signalling

[55] Norma IEC 62912: Railway applications. Direct current signalling monostable relays of type N and type C

[56] Norma IEC 62912-2:2019 Railway applications. Direct current signalling monostable relays. Part 2: Spring type relays

[57] Norma PN-EN 50124-1:2017-09 Zastosowania kolejowe. Koordynacja izolacji. Część 1: Wymagania podstawowe. Odstępy izolacyjne powietrzne i powierzchniowe dla całego wyposażenia elektrycznego i elektronicznego

[58] Norma PN-EN 50578:2014-02 Zastosowania kolejowe. Przekaźniki sygnalizacyjne prądu stałego

[59] Norma PN-EN 61810-3 Elektromechaniczne przekaźniki pośredniczące. Część 3: Przekaźniki z zestykami o wymuszonym przełączaniu (mechanicznym sprzężeniu)

[60] Norma PN-EN IEC 60812:2018-12 Analiza rodzajów i skutków uszkodzeń (FMEA i FMECA)

[61] Norma UIC 736 : 4ED 2004 Signalling relays

[62] Nowa encyklopedia powszechna PWN. Tom 3. Wydawnictwo Naukowe PWN, Warszawa 1995

[63] Reniers M., Willemse T.: Folk Theorems on the Correspondence between State-Based and Event-Based Systems. In: Proceedings of the 37th Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Slovakia, 22–28 January 2011; pp. 494–505, doi:10.1007/978-3-642-18381-2_41

[64] Rozporządzenie Ministra Infrastruktury i Rozwoju z dnia 13 maja 2014 r. w sprawie dopuszczania do eksploatacji określonych rodzajów budowli, urządzeń i pojazdów kolejowych, Dz.U. 2014 poz. 720 z późn. zm.

[65] Rozporządzenie Ministra Infrastruktury z dnia 18 lipca 2005 r. w sprawie ogólnych warunków prowadzenia ruchu kolejowego i sygnalizacji, Dz. U. Nr 172 poz. 1444, z późn. zm.

[66] Szydłowski J., Jasiński S., Grochowski K.: WT PW / DTR WT UZ / 01 / 07 Dokumentacja Techniczno-Ruchowa elektronicznych urządzeń nastawczych z komputerowymi

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

urządzeniami zależnościowymi typu WT UZ, Politechnika Warszawska, Wydział Transportu, Zakład Sterowania Ruchem Kolejowym. Warszawa 2007

[67] Tai K.: Definitions and Detection of Deadlock, Livelock, and Starvation in Concurrent Programs. In: 1994 International Conference on Parallel Processing (ICPP'94), Raleigh, NC, 15–19 August 1994; Agrawal, D.P., Ed.; CRC Press: Boca Raton, FL, USA, 1994; pp. 69–72

[68] Tanenbaum A., Wetherall D.: Sieci komputerowe. Wydanie 5. Helion. Gliwice, 2012

[69] Theeg G., Vlasenko S.: Railway Signalling & Interlocking International Compendium. Eurailpress, Hamburg 2009

[70] Ustawa z dnia 28 marca 2003 r. o transporcie kolejowym, Dz. U. z 2020 r. poz. 1043 z późn. zm.

[71] Warunki bezpiecznej instalacji i eksploatacji urządzeń sterowania ruchem kolejowym na liniach kolejowych zarządzanych przez PKP Polskie Linie Kolejowe S.A. Ie-100a. Warszawa 2023

[72] Wawrzyński W.: Podstawy współczesnej elektroniki. OWPW, Warszawa 2003

[73] Wheatcraft L.: Everything you wanted to know about interfaces, but were afraid to ask. Requirements Experts, 2010

[74] Wontorski P., Dzierżak M.: Digital standard interfaces in railway traffic control systems. W: WUT Journal of Transportation Engineering, vol. 131 2020, s. 17-29.

[75] Wontorski P., Dzierżak M.: Modułowa standardowa architektura systemów kierowania i sterowania ruchem kolejowym–aktualne trendy. W: Zeszyty Naukowo-Techniczne Stowarzyszenia Inżynierów i Techników Komunikacji w Krakowie. Seria: Materiały Konferencyjne, nr 2(123), Kraków 2021

[76] Wontorski P., Kochan A.: Komputerowe systemy kierowania i sterowania ruchem kolejowym. Część 1: Funkcje, elementy i układy, OWPW, Warszawa 2020

[77] Wontorski P.: Wybrane aspekty projektowania interfejsów i powiązań urządzeń sterowania ruchem kolejowym, prezentacja podczas: XVIII Konferencja Naukowo-Techniczna: Nowoczesne technologie i systemy zarządzania w transporcie szynowym, Zakopane, 04-06.12.2019. Dostęp on-line 03.04.2023: https://www.researchgate.net/publication/346734427_Wybrane_aspekty_projektowania_interfejsow_i_powiazan_urzadzen_sterowania_ruchem_kolejowym

[78] Wymagania bezpieczeństwa dla urządzeń sterowania ruchem kolejowym. Zakład Sterowania Ruchem i Zasilania CNTK, Warszawa, 1997

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

[79] Wytyczne techniczne budowy urządzeń sterowania ruchem kolejowym Ie-4 (WTB-E10). PKP PLK S.A, Warszawa 2019

[80] Zabłocki W.: Modelowanie stacyjnych systemów sterowania ruchem kolejowym. Prace Naukowe Politechniki Warszawskiej, z. 65. Transport. OWPW, Warszawa 2008

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

Załącznik 1. Model układu oddziaływania pociągu w języku IMDS przed użyciem makrogeneratora

```
// ##### Crossbar 1 ##########
//      Port: [W:6] In  Vcc     Relay \ Contact2 normally open
//      Port: [Z:2] In  Vcc     Relay \ Contact2 normally open
//      Port: [dKo:6] In  Vcc   Relay \ Contact2 normally open
//      Port: [it:2] In  Vcc    Relay \ Contact2 normally open
//      Port: [it:12] In  Vcc   Relay \ Contact2 normally closed
//      Port: [:+24V] Out  Vcc  Power supply \ Power supply
//      Port: [ldKo:22] In  Vcc         Relay \ Coil
//        Port: [pdKo:14] In  Vcc           Switch2 normally open \
Contact2 normally open
#INCLUDE Fork.txt parameters(7, "v", 11, 1)

// ##### Crossbar 2 ##########
//      Port: [Dp:11] In  GND   Relay \ Contact2 normally closed
//      Port: [:-24V] Out  GND  Power supply \ Power supply

// ##### Crossbar 3 ##########
//      Port: [dKo:9] In  GND   Relay \ Contact2 normally open
//      Port: [dKo:1] In  GND   Relay \ Contact2 normally open
//      Port: [dKo:21] In  GND  Relay \ Coil
//      Port: [W:1] In  GND     Relay \ Contact2 normally open
//      Port: [Dp:12] Out  GND  Relay \ Contact2 normally closed
//      Port: [Kc:11] In  GND   Relay \ Contact2 normally closed
#INCLUDE Fork.txt parameters(5, "g", 11, 3)

// ##### Crossbar 4 ##########
//        Port: [pdKo:13] Out   Vcc          Switch2 normally open \
Contact2 normally open
//      Port: [dKo:5] Out  Vcc  Relay \ Contact2 normally open
//      Port: [dKo:22] In  Vcc  Relay \ Coil
#INCLUDE Join.txt parameters(2, "v", 11, 4)

// ##### Crossbar 5 ##########
//      Port: [ldKo:21] In  GND         Relay \ Coil
//     Port: [dKo:10] Out  GND        Relay \ Contact2 normally open

// ##### Crossbar 6 ##########
//      Port: [Kpz:2] Out  GND  Relay \ Contact2 normally open
//      Port: [W:2] Out  GND    Relay \ Contact2 normally open
//      Port: [W:3] In  GND     Relay \ Contact2 normally open
//      Port: [W:21] In  GND    Relay \ Coil
//      Port: [dKo:2] Out  GND  Relay \ Contact2 normally open
#INCLUDE Join.txt parameters(3, "g", 11, 6)
#INCLUDE Fork.txt parameters(2, "g", 11, 6)

// ##### Crossbar 7 ##########
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
//      Port: [Z:12] Out  Vcc   Relay \ Contact2 normally closed
//      Port: [W:5] Out  Vcc    Relay \ Contact2 normally open
//      Port: [W:22] In  Vcc    Relay \ Coil
#INCLUDE Join.txt parameters(2, "v", 11, 7)

// ##### Crossbar 8 ##########
//      Port: [Jz:1] Out  Vcc   Relay \ Contact2 normally open
//      Port: [Z:11] In  Vcc    Relay \ Contact2 normally closed

// ##### Crossbar 9 ##########
//      Port: [Jz:2] In  Vcc    Relay \ Contact2 normally open
//      Port: [Jt:11] Out  Vcc  Relay \ Contact2 normally closed

// ##### Crossbar 10 ##########
//      Port: [Jt:12] In  Vcc   Relay \ Contact2 normally closed
//      Port: [it:11] Out  Vcc  Relay \ Contact2 normally closed

// ##### Crossbar 11 ##########
//      Port: [Kc:12] Out  GND  Relay \ Contact2 normally closed
//      Port: [Kpz:1] In  GND   Relay \ Contact2 normally open

// ##### Crossbar 12 ##########
//      Port: [Z:22] In  Vcc    Relay \ Coil
//      Port: [Z:1] Out  Vcc    Relay \ Contact2 normally open
//      Port: [Jz:11] Out  Vcc  Relay \ Contact2 normally closed
#INCLUDE Join.txt parameters(2, "v", 11, 12)

// ##### Crossbar 13 ##########
//      Port: [Jt:1] Out  Vcc   Relay \ Contact2 normally open
//      Port: [Jz:12] In  Vcc   Relay \ Contact2 normally closed

// ##### Crossbar 14 ##########
//      Port: [Jt:2] In  Vcc    Relay \ Contact2 normally open
//      Port: [it:1] Out  Vcc   Relay \ Contact2 normally open

// ##### Crossbar 15 ##########
//      Port: [W:4] Out  GND    Relay \ Contact2 normally open
//      Port: [Z:21] In  GND    Relay \ Coil

// ##### Contact 1 - Contact2 normally open ##########
//      Port: [dKo:1] In  GND   Relay \ Contact2 normally open
//      Port: [dKo:2] Out  GND  Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(3, "g", 11, 1)

// ##### Contact 2 - Contact2 normally open ##########
//      Port: [dKo:5] Out  Vcc  Relay \ Contact2 normally open
//      Port: [dKo:6] In  Vcc   Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(2, "v", 11, 2)
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
// ##### Contact 3 - Contact2 normally open ##########
//      Port: [dKo:9] In  GND   Relay \ Contact2 normally open
//      Port: [dKo:10] Out  GND Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(1, "g", 11, 3)

// ##### Contact 4 - Contact2 normally closed ##########
//      Port: [Dp:11] In  GND   Relay \ Contact2 normally closed
//      Port: [Dp:12] Out  GND  Relay \ Contact2 normally closed
#INCLUDE Contact.txt parameters(1, "g", 11, 4)

// ##### Contact 5 - Contact2 normally open ##########
//      Port: [it:1] Out  Vcc   Relay \ Contact2 normally open
//      Port: [it:2] In  Vcc    Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(1, "v", 11, 5)

// ##### Contact 6 - Contact2 normally closed ##########
//      Port: [it:11] Out  Vcc  Relay \ Contact2 normally closed
//      Port: [it:12] In  Vcc   Relay \ Contact2 normally closed
#INCLUDE Contact.txt parameters(1, "v", 11, 6)

// ##### Contact 7 - Contact2 normally open ##########
//      Port: [Jt:1] Out  Vcc   Relay \ Contact2 normally open
//      Port: [Jt:2] In  Vcc    Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(1, "v", 11, 7)

// ##### Contact 8 - Contact2 normally closed ##########
//      Port: [Jt:11] Out  Vcc  Relay \ Contact2 normally closed
//      Port: [Jt:12] In  Vcc   Relay \ Contact2 normally closed
#INCLUDE Contact.txt parameters(1, "v", 11, 8)

// ##### Contact 9 - Contact2 normally open ##########
//      Port: [Jz:1] Out  Vcc   Relay \ Contact2 normally open
//      Port: [Jz:2] In  Vcc    Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(1, "v", 11, 9)

// ##### Contact 10 - Contact2 normally closed ##########
//      Port: [Jz:11] Out  Vcc  Relay \ Contact2 normally closed
//      Port: [Jz:12] In  Vcc   Relay \ Contact2 normally closed
#INCLUDE Contact.txt parameters(2, "v", 11, 10)

// ##### Contact 11 - Contact2 normally closed ##########
//      Port: [Kc:11] In  GND   Relay \ Contact2 normally closed
//      Port: [Kc:12] Out  GND  Relay \ Contact2 normally closed
#INCLUDE Contact.txt parameters(1, "g", 11, 11)

// ##### Contact 12 - Contact2 normally open ##########
//      Port: [Kpz:1] In  GND   Relay \ Contact2 normally open
//      Port: [Kpz:2] Out  GND  Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(1, "g", 11, 12)
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
// ##### Contact 13 - Contact2 normally open ##########
//       Port: [pdKo:13] Out  Vcc          Switch2 normally open \
Contact2 normally open
//       Port: [pdKo:14] In  Vcc Switch2 normally open \ Contact2
normally open
#INCLUDE Contact.txt parameters(1, "v", 11, 13)

// ##### Contact 14 - Contact2 normally open ##########
//      Port: [W:1] In  GND     Relay \ Contact2 normally open
//      Port: [W:2] Out  GND    Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(2, "g", 11, 14)

// ##### Contact 15 - Contact2 normally open ##########
//      Port: [W:3] In  GND     Relay \ Contact2 normally open
//      Port: [W:4] Out  GND    Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(1, "g", 11, 15)

// ##### Contact 16 - Contact2 normally open ##########
//      Port: [W:5] Out  Vcc    Relay \ Contact2 normally open
//      Port: [W:6] In  Vcc     Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(2, "v", 11, 16)

// ##### Contact 17 - Contact2 normally open ##########
//      Port: [Z:1] Out  Vcc    Relay \ Contact2 normally open
//      Port: [Z:2] In  Vcc     Relay \ Contact2 normally open
#INCLUDE Contact.txt parameters(1, "v", 11, 17)

// ##### Contact 18 - Contact2 normally closed ##########
//      Port: [Z:11] In  Vcc    Relay \ Contact2 normally closed
//      Port: [Z:12] Out  Vcc   Relay \ Contact2 normally closed
#INCLUDE Contact.txt parameters(1, "v", 11, 18)

// ##### Coil 1 - Coil ##########
//      Port: [dKo:21] In  GND  Relay \ Coil
//      Port: [dKo:22] In  Vcc  Relay \ Coil
#INCLUDE Coil.txt parameters("g", "v", 11, 1)

// ##### Coil 2 - Coil ##########
//      Port: [ldKo:21] In  GND Relay \ Coil
//      Port: [ldKo:22] In  Vcc Relay \ Coil
#INCLUDE Coil.txt parameters("g", "v", 11, 2)

// ##### Coil 3 - Coil ##########
//      Port: [W:21] In  GND    Relay \ Coil
//      Port: [W:22] In  Vcc    Relay \ Coil
#INCLUDE Coil.txt parameters("g", "v", 11, 3)

// ##### Coil 4 - Coil ##########
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
//      Port: [Z:21] In  GND    Relay \ Coil
//      Port: [Z:22] In  Vcc    Relay \ Coil
#INCLUDE Coil.txt parameters("g", "v", 11, 4)

// ##### Relay anchor 1 ##########
//      Relay: [dKo]  Relay
#INCLUDE Arm_NO.txt parameters(3 ,2, 11, 1)

// ##### Relay anchor 2 ##########
//      Relay: [Dp]  Relay
#INCLUDE Arm2_NC.txt parameters(1 ,3, 11, 2)

// ##### Relay anchor 3 ##########
//      Relay: [it]  Relay
#INCLUDE Arm_NONC.txt parameters(1 ,1 ,4, 11, 3)

// ##### Relay anchor 4 ##########
//      Relay: [Jt]  Relay
#INCLUDE Arm_NONC.txt parameters(1 ,1 ,5, 11, 4)

// ##### Relay anchor 5 ##########
//      Relay: [Jz]  Relay
#INCLUDE Arm_NONC.txt parameters(1 ,1 ,6, 11, 5)

// ##### Relay anchor 6 ##########
//      Relay: [Kc]  Relay
#INCLUDE Arm_NC.txt parameters(1 ,7, 11, 6)

// ##### Relay anchor 7 ##########
//      Relay: [Kpz]  Relay
#INCLUDE Arm_NO.txt parameters(1 ,8, 11, 7)

// ##### Relay anchor 8 ##########
//      Relay: [ldKo]  Relay
#INCLUDE Arm_0.txt parameters(9, 11, 8)

// ##### Relay anchor 9 ##########
//      Relay: [W]  Relay
#INCLUDE Arm_NO.txt parameters(3 ,10, 11, 9)

// ##### Relay anchor 10 ##########
//      Relay: [Z]  Relay
#INCLUDE Arm_NONC.txt parameters(1 ,1 ,11, 11, 10)

servers
Cont1_dKo:cont1,  Cont2_dKo:cont2,  Cont3_dKo:cont3,  Cont4_Dp:cont4,
Cont5_it:cont5,   Cont6_it:cont6,   Cont7_Jt:cont7,   Cont8_Jt:cont8,
Cont9_Jz:cont9,          Cont10_Jz:cont10,          Cont11_Kc:cont11,
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
Cont12_Kpz:cont12,        Cont13_pdKo:cont13,        Cont14_W:cont14,
Cont15_W:cont15, Cont16_W:cont16, Cont17_Z:cont17, Cont18_Z:cont18,
Coil1_dKo:coil1, Coil2_ldKo:coil2, Coil3_W:coil3, Coil4_Z:coil4,
Arm1_dKo:arm1,     Arm2_Dp:arm2,     Arm3_it:arm3,     Arm4_Jt:arm4,
Arm5_Jz:arm5,     Arm6_Kc:arm6,     Arm7_Kpz:arm7,     Arm8_ldKo:arm8,
Arm9_W:arm9, Arm10_Z:arm10,
Fork1:fork1,  Fork3:fork3,  Join4:join4,  Join6:join6,  Fork6:fork6,
Join7:join7, Join12:join12,
Env;

agents
AEnv, A_dKo, A_Dp, A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z;

init ->
{
        Cont1_dKo(Arm1_dKo, Fork3, Join6, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont2_dKo(Arm1_dKo, Fork1, Join4, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont3_dKo(Arm1_dKo, Fork3, Coil2_ldKo, AEnv, A_dKo, A_Dp,
A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont4_Dp(Arm2_Dp, Env, Fork3, AEnv, A_dKo, A_Dp, A_it, A_Jt,
A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_onm0,
        Cont5_it(Arm3_it, Fork1, Cont7_Jt, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont6_it(Arm3_it, Fork1, Cont8_Jt, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_onm0,
        Cont7_Jt(Arm4_Jt, Cont5_it, Cont10_Jz, AEnv, A_dKo, A_Dp,
A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont8_Jt(Arm4_Jt, Cont6_it, Cont9_Jz, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_onm0,
        Cont9_Jz(Arm5_Jz, Cont8_Jt, Cont18_Z, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont10_Jz(Arm5_Jz, Cont7_Jt, Join12, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_onm0,
        Cont11_Kc(Arm6_Kc, Fork3, Cont12_Kpz, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_onm0,
        Cont12_Kpz(Arm7_Kpz, Cont11_Kc, Join6, AEnv, A_dKo, A_Dp,
A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont13_pdKo(Env, Fork1, Join4, AEnv, A_dKo, A_Dp, A_it, A_Jt,
A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont14_W(Arm9_W, Fork3, Join6, AEnv, A_dKo, A_Dp, A_it, A_Jt,
A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont15_W(Arm9_W, Fork6, Coil4_Z, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont16_W(Arm9_W, Fork1, Join7, AEnv, A_dKo, A_Dp, A_it, A_Jt,
A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
        Cont17_Z(Arm10_Z, Fork1, Join12, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_ofm0,
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
        Cont18_Z(Arm10_Z, Cont9_Jz, Join7, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).S0_onm0,
        Coil1_dKo(Fork3, Join4, Arm1_dKo, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Uof_c0b0,
        Coil2_ldKo(Cont3_dKo, Fork1, Arm8_ldKo, AEnv, A_dKo, A_Dp,
A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Uof_c0b0,
        Coil3_W(Fork6, Join7, Arm9_W, AEnv, A_dKo, A_Dp, A_it, A_Jt,
A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Uof_c0b0,
        Coil4_Z(Cont15_W, Join12, Arm10_Z, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Uof_c0b0,

        Arm1_dKo(Coil1_dKo, Cont1_dKo, Cont2_dKo, Cont3_dKo, AEnv,
A_dKo, A_Dp, A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Arm2_Dp(Env, Env, Cont4_Dp, AEnv, A_dKo, A_Dp, A_it, A_Jt,
A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof_TofUof,
        Arm3_it(Env, Cont5_it, Cont6_it, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Arm4_Jt(Env, Cont7_Jt, Cont8_Jt, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Arm5_Jz(Env, Cont9_Jz, Cont10_Jz, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Arm6_Kc(Env, Cont11_Kc, AEnv, A_dKo, A_Dp, A_it, A_Jt, A_Jz,
A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Arm7_Kpz(Env, Cont12_Kpz, AEnv, A_dKo, A_Dp, A_it, A_Jt, A_Jz,
A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Arm8_ldKo(Coil2_ldKo, AEnv, A_dKo, A_Dp, A_it, A_Jt, A_Jz,
A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Arm9_W(Coil3_W, Cont14_W, Cont15_W, Cont16_W, AEnv, A_dKo,
A_Dp, A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Arm10_Z(Coil4_Z, Cont17_Z, Cont18_Z, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).Sof,
        Fork1(Env, Cont16_W, Cont17_Z, Cont2_dKo, Cont5_it, Cont6_it,
Coil2_ldKo, Cont13_pdKo, AEnv, A_dKo, A_Dp, A_it, A_Jt, A_Jz, A_Kc,
A_Kpz, A_ldKo, A_W, A_Z).X0,
        Fork3(Cont4_Dp, Cont3_dKo, Cont1_dKo, Coil1_dKo, Cont14_W,
Cont11_Kc, AEnv, A_dKo, A_Dp, A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo,
A_W, A_Z).X0,
        Join4(Cont13_pdKo, Cont2_dKo, Coil1_dKo, AEnv, A_dKo, A_Dp,
A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).F0_00,
        Join6(Cont12_Kpz, Cont14_W, Cont1_dKo, Fork6, AEnv, A_dKo,
A_Dp, A_it, A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).F0_000,
        Fork6(Join6, Cont15_W, Coil3_W, AEnv, A_dKo, A_Dp, A_it, A_Jt,
A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).X0,
        Join7(Cont18_Z, Cont16_W, Coil3_W, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).F0_00,
        Join12(Cont17_Z, Cont10_Jz, Coil4_Z, AEnv, A_dKo, A_Dp, A_it,
A_Jt, A_Jz, A_Kc, A_Kpz, A_ldKo, A_W, A_Z).F0_00,
        Env().start,
```


Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21

```
AEnv.Env.ini, A_dKo.Arm1_dKo.idl, A_Dp.Arm2_Dp.idl, A_it.Arm3_it.idl,
A_Jt.Arm4_Jt.idl,         A_Jz.Arm5_Jz.idl,         A_Kc.Arm6_Kc.idl,
A_Kpz.Arm7_Kpz.idl,       A_ldKo.Arm8_ldKo.idl,       A_W.Arm9_W.idl,
A_Z.Arm10_Z.idl
}
```

Pobrano z https://repo.pw.edu.pl / Downloaded from Repository of Warsaw University of Technology 2025-03-21